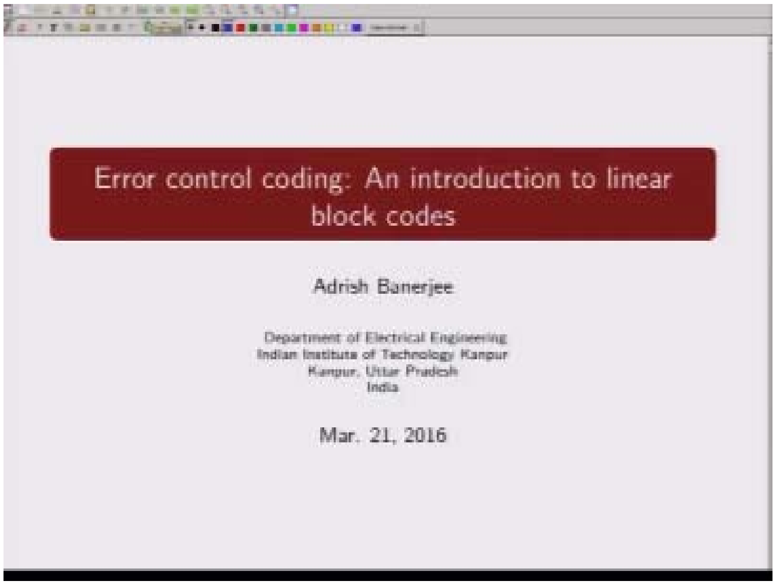
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture-5C
Distance Properties of Linear Block Codes-II

by
Dr. Adrish Banerjee
Department of Electrical Engineering, IIT Kanpur

Welcome to the course on error control coding, an introduction to linear block codes.

(Refer Slide Time: 00:19)



Error control coding: An introduction to linear block codes

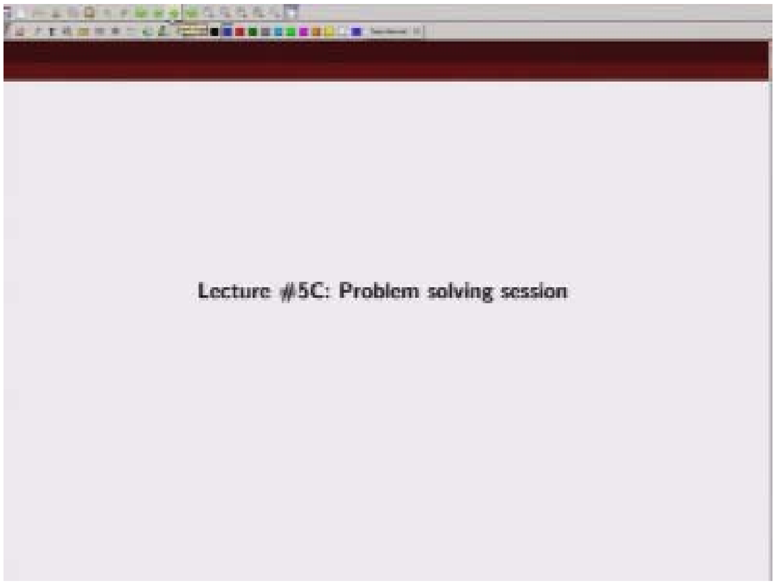
Adrish Banerjee

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh
India

Mar. 21, 2016

So, so far we have studied what are linear block codes, how do we describe linear block codes using generator matrix and parity check matrix. We talked about how we can use error correcting codes for error detection and error correction and we discussed the distance properties of linear block codes. Today we will spend some time solving some problems for linear block codes.

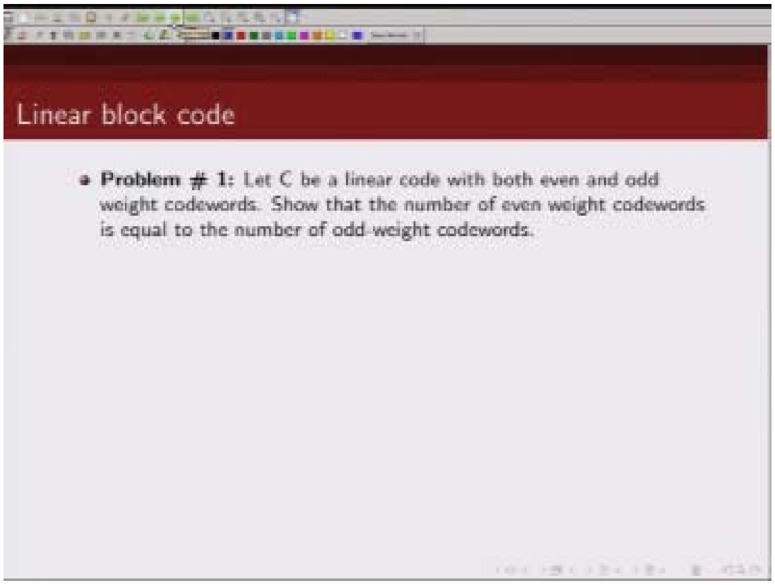
(Refer Slide Time: 00:51)



Lecture #5C: Problem solving session

So today's session will be on problem solving.

(Refer Slide Time: 00:57)

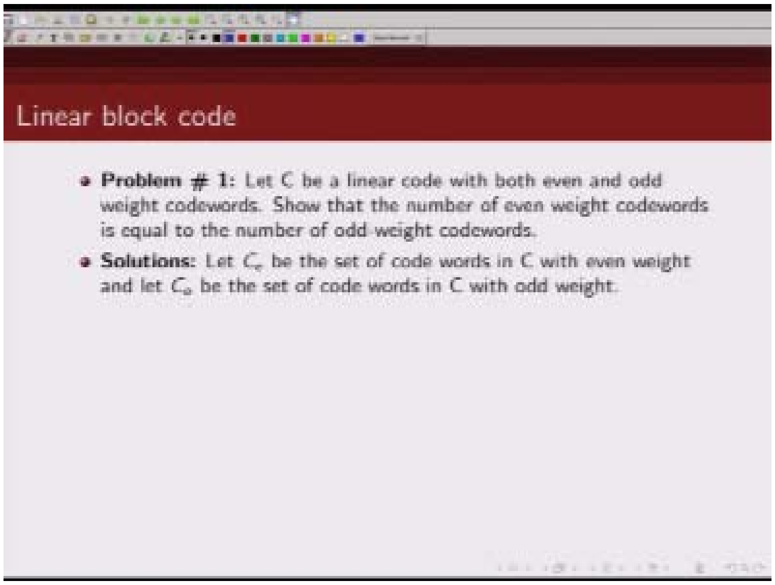


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd weight codewords.

So the first problem that we will look at is let C be a linear code with both even and odd weight codewords. Prove that the number of even bit codewords is equal to number of odd weight codewords.

(Refer Slide Time: 01:10)

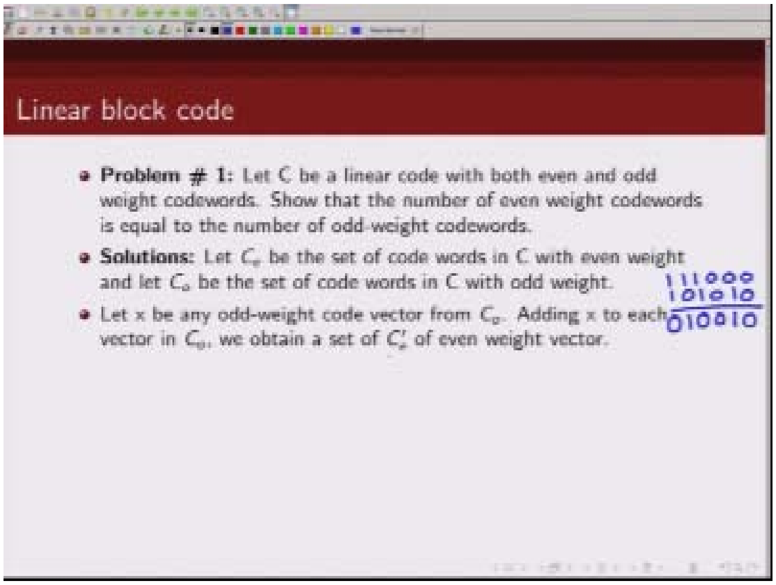


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.

So let us denote the set of even code words in C by C_e and set of odd code words in C by C_o.

(Refer Slide Time: 01:26)



Linear block code

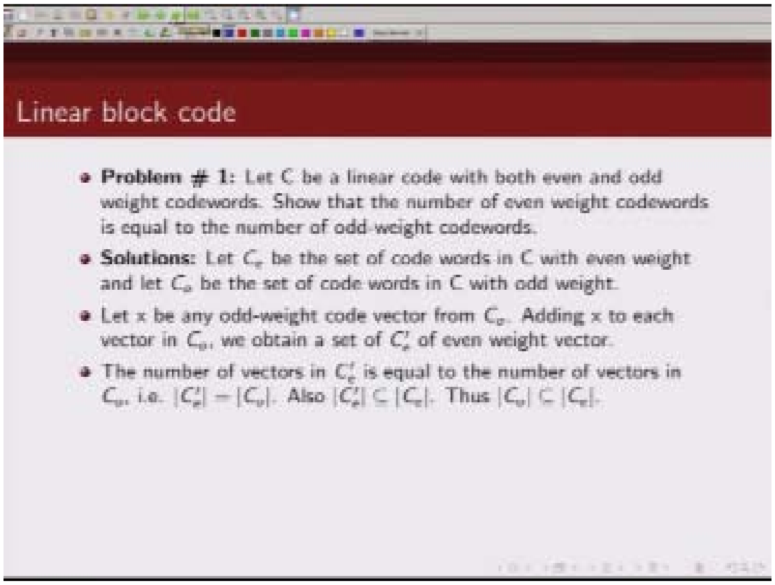
- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.

$$\begin{array}{r} 111000 \\ 101010 \\ \hline 010010 \end{array}$$

Now let us consider an odd weight code word x which is taken from the set C_o, and let us add x to each of the code words which are there in the set C_o. So if we add a odd weight code word to another odd weight code word what we will get is a even weight code word. For example, let us say I add 111000 and I add 101010 so this first code word, this is odd weight code word, its weight is 3, similarly this code word also has weight 3.

If I add both of them what do I get, I get 010010 and this is a even weight code word. So when I add x which is an odd weight code vector and I add x to each of the elements in this set C_o what I get is a set of even code words vectors. And let us denote that set by C_e'.

(Refer Slide Time: 02:58)

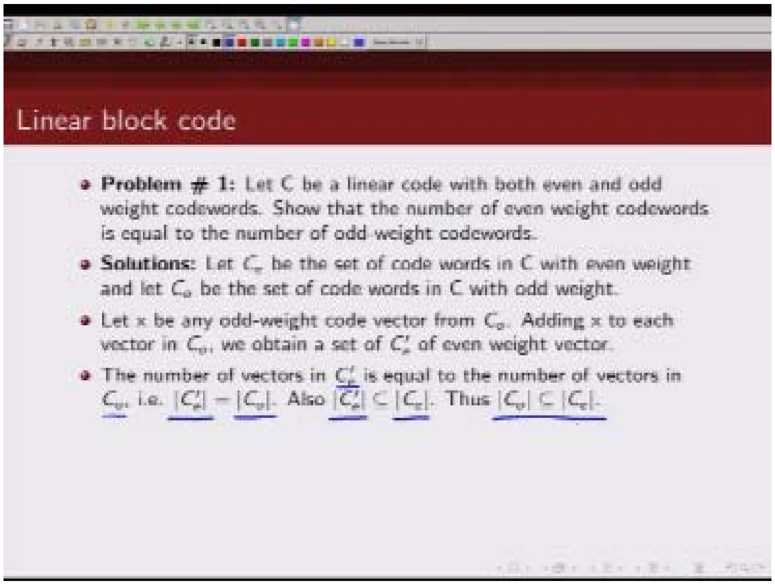


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subseteq |C_e|$. Thus $|C_o| \subseteq |C_e|$.

Now the number of code vectors in C_e' is going to be equal to number of vectors in C_o, why, because how did we get this C_e'? We added an odd vector x to the set C_o.

(Refer Slide Time: 03:19)

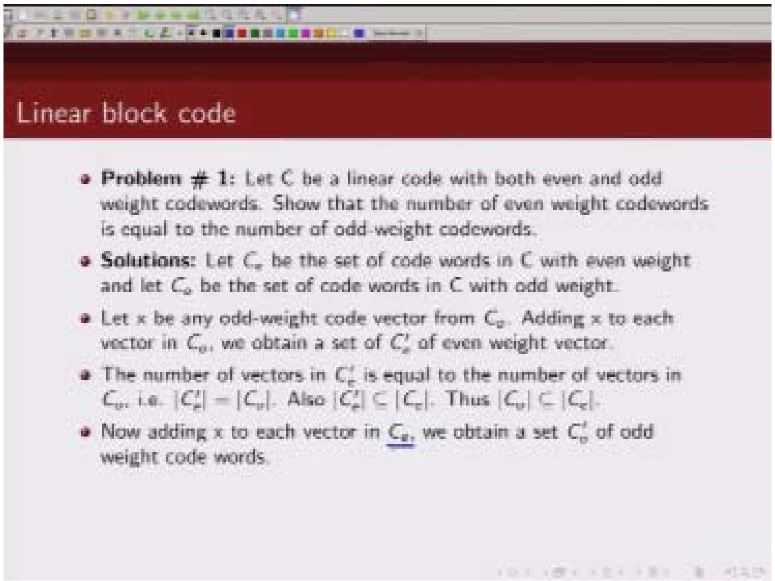


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subset |C_e|$. Thus $|C_o| \subset |C_e|$.

So number of vectors in this set is going to be equal to number of vectors in C_o. Hence number of elements in C_e' is going to be same as number of elements in C_o and since we know that this set of even vectors C_e' is the subset of set of even vectors. We can write from this that number of elements in the set of number of odd code words is going to be a subset of number of even code words.

(Refer Slide Time: 04:08)

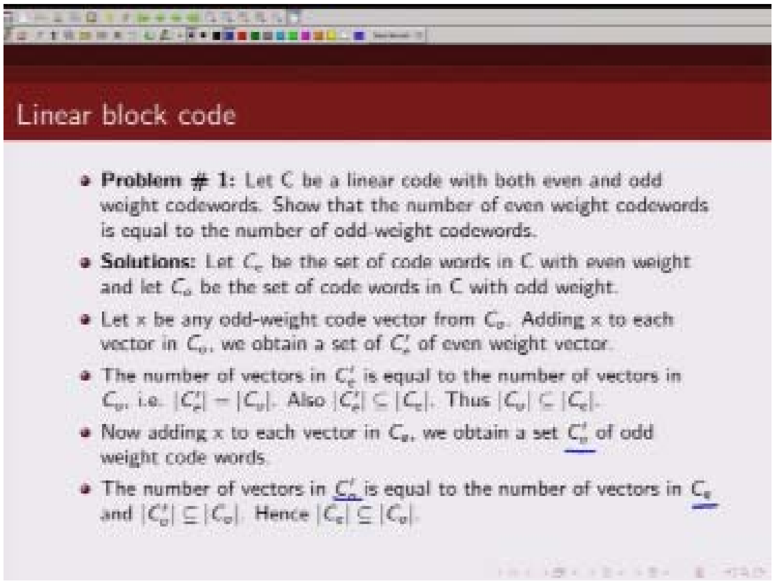


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subseteq |C_e|$. Thus $|C_o| \subseteq |C_e|$.
- Now adding x to each vector in $\underline{C_e}$, we obtain a set C_o' of odd weight code words.

Next let us add the same odd weight code word now to all the vectors in the set C_e. So if we add an odd weight code word to set off even weight code words what we will get is a set of odd code words.

(Refer Slide Time: 04:34)

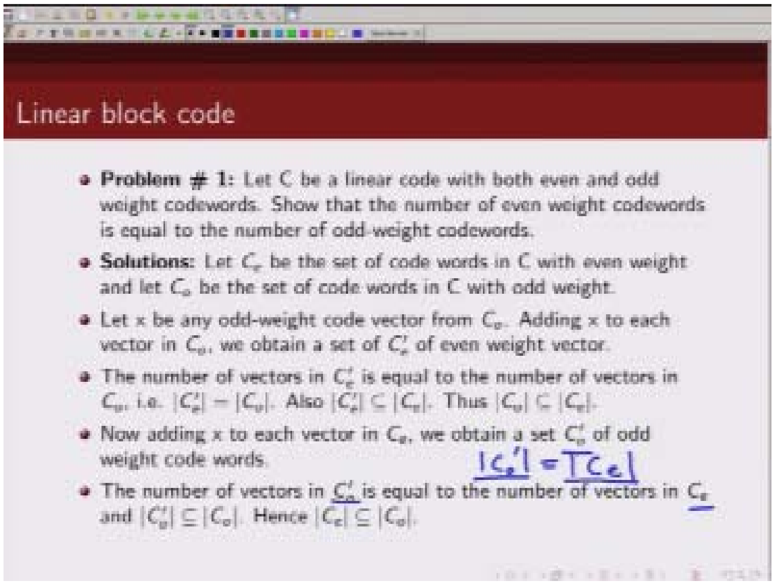


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subseteq |C_e|$. Thus $|C_o| \subseteq |C_e|$.
- Now adding x to each vector in C_e, we obtain a set $\underline{C_o'}$ of odd weight code words.
- The number of vectors in $\underline{C_o'}$ is equal to the number of vectors in $\underline{C_e}$ and $|C_o'| \subseteq |C_o|$. Hence $|C_e| \subseteq |C_o|$.

So the number of vectors in Co′ is going to be equal to number of vectors in number of even vectors. Why? Because this set was generated by adding an odd vector x to the set of even code words.

(Refer Slide Time: 05:03)

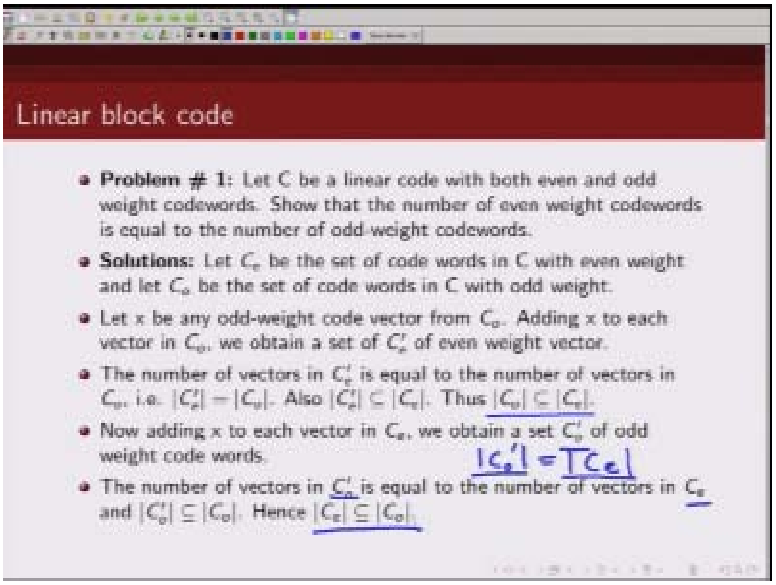


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subseteq |C_e|$. Thus $|C_o| \subseteq |C_e|$.
- Now adding x to each vector in C_e, we obtain a set C_o' of odd weight code words.
  $\underline{|C_o'|} = \underline{|C_e|}$
- The number of vectors in $\underline{C_o'}$ is equal to the number of vectors in $\underline{C_e}$ and $|C_o'| \subseteq |C_o|$. Hence $|C_e| \subseteq |C_o|$.

So we can then write that C_o' is equal to this okay. The set of code words here is same as set of code words here. Now we know that C_{-}' is a subset of set of odd code words. So then from this relation and this relation we can write that set of even code words is a subset of set of number of elements in this is a subset of number of is basically less than number of elements in this set.

(Refer Slide Time: 05:48)

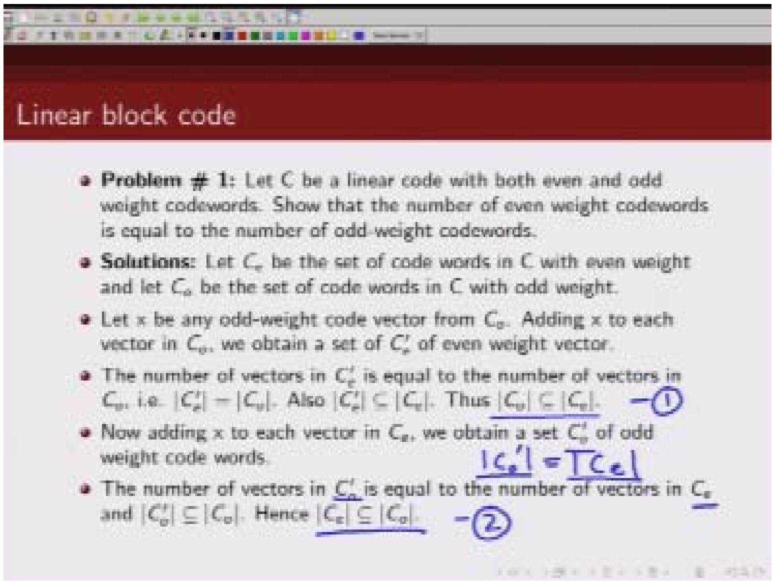


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subseteq |C_e|$. Thus $\underline{|C_o| \subseteq |C_e|}$.
- Now adding x to each vector in C_e, we obtain a set C_o' of odd weight code words. $\underline{|C_o'|} = \underline{|C_e|}$
- The number of vectors in $\underline{C_o'}$ is equal to the number of vectors in $\underline{C_e}$ and $|C_o'| \subseteq |C_o|$. Hence $\underline{|C_e| \subseteq |C_o|}$.

Now from this relation and this relation both of them can be true only if number of elements in C₀ is same as number of elements in Ce

(Refer Slide Time: 05:59)

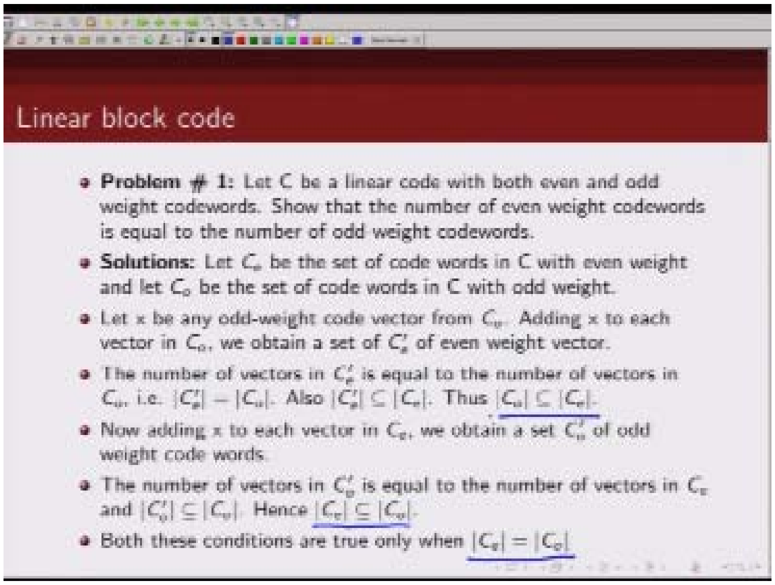


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subseteq |C_e|$. Thus $\underline{|C_o| \subseteq |C_e|}$. $-①$
- Now adding x to each vector in C_e, we obtain a set C_o' of odd weight code words. $\underline{|C_o'|} = \underline{|C_e|}$
- The number of vectors in $\underline{C_o'}$ is equal to the number of vectors in $\underline{C_e}$ and $|C_o'| \subseteq |C_o|$. Hence $\underline{|C_e| \subseteq |C_o|}$. $-②$

So this relation let us call it 1 and let us call it 2. These two relations are satisfied only if we have set of even code words to be same as set of odd code words. Hence we prove that in a linear code with both even and odd code words, the number of even weight code words is same as number of odd weight code words.

(Refer Slide Time: 06:31)

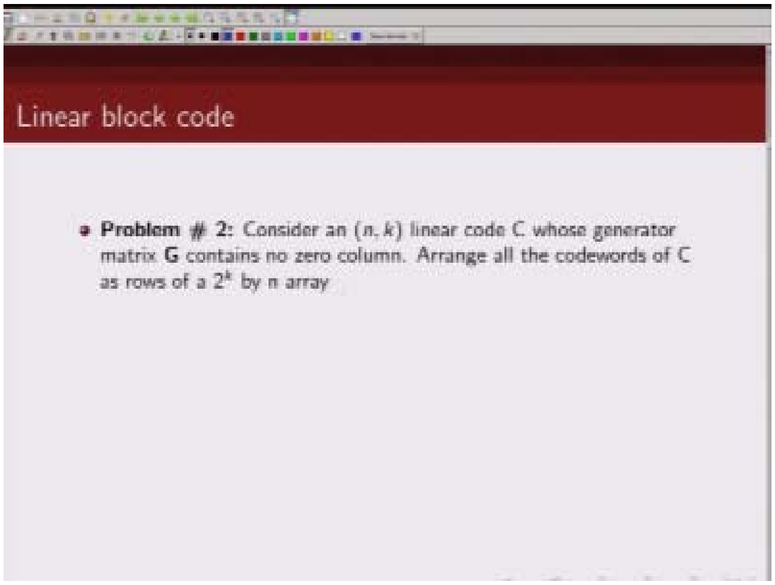


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o. Adding x to each vector in C_o, we obtain a set of C_e' of even weight vector.
- The number of vectors in C_e' is equal to the number of vectors in C_o, i.e. $|C_e'| = |C_o|$. Also $|C_e'| \subseteq |C_e|$. Thus $\underline{|C_o| \subseteq |C_e|}$.
- Now adding x to each vector in C_e, we obtain a set C_o' of odd weight code words.
- The number of vectors in C_o' is equal to the number of vectors in C_e and $|C_o'| \subseteq |C_o|$. Hence $\underline{|C_e| \subseteq |C_o|}$.
- Both these conditions are true only when $\underline{|C_e| = |C_o|}$

So I repeat, this condition and this condition will be simultaneously satisfied only when the set of number of even code words is same as set of number of odd code words. And this proves our result.

(Refer Slide Time: 06:54)

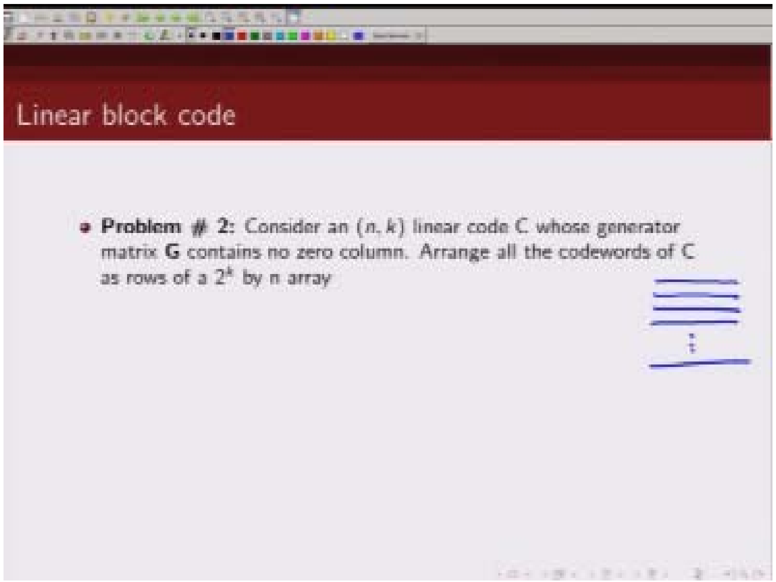


Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

The next problem that we will look at is as follows. Let us consider a linear (n, k) code C whose generator matrix contains no zero column. Now arrange all code words of this linear code C as rows of $2^k/n$ array.

(Refer Slide Time: 07:22)

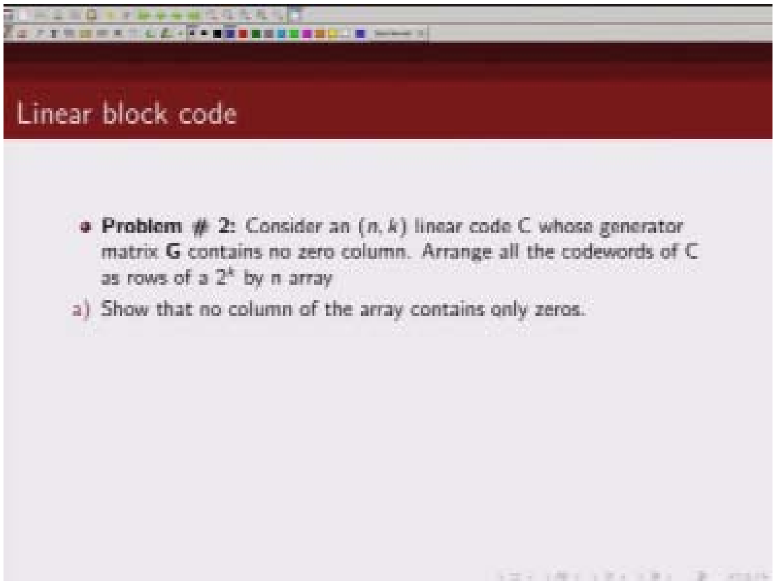


Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

So what we are doing is we are arranging the 2^k code words like this in an array. So this array has dimension 2^kxn because total number of code words are 2^k for a (n, k) binary code and they are all n bit.

(Refer Slide Time: 07:43)



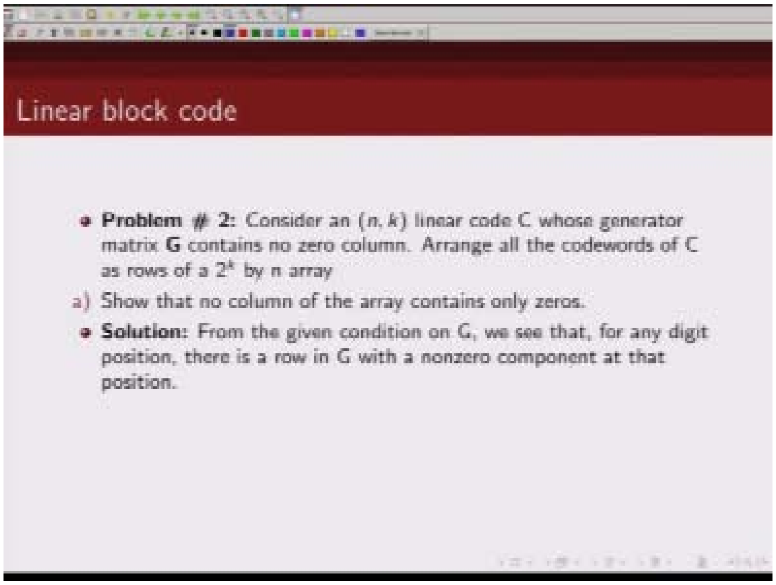
Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix $\mathbf{G}$ contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

a) Show that no column of the array contains only zeros.

The first result that we are going to show is no columns of this array contains zero.

(Refer Slide Time: 07:52)



Linear block code

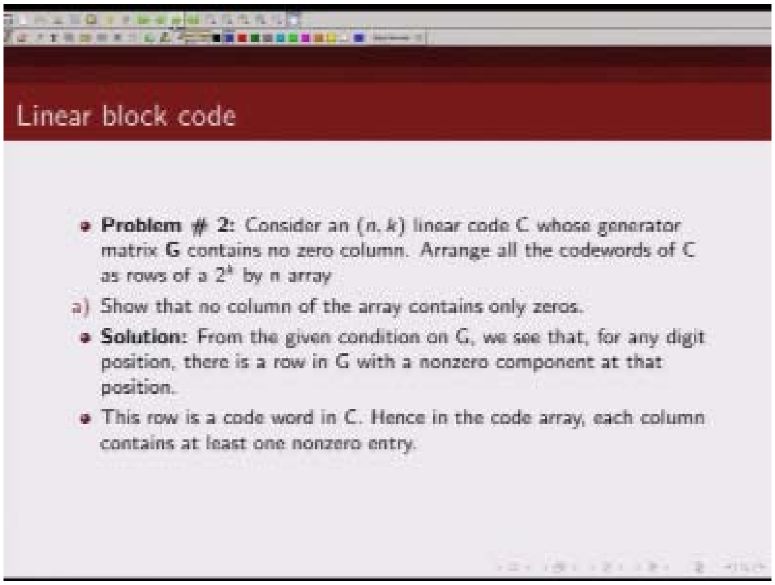
- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

a) Show that no column of the array contains only zeros.

- **Solution:** From the given condition on G, we see that, for any digit position, there is a row in G with a nonzero component at that position.

Now please note that we have been given that the generator matrix G does not contain any zero column okay. So from the given condition on G we can see that for any position of any bit position there is a row in G which has a non zero component at that particular bit location.

(Refer Slide Time: 08:21)



Linear block code

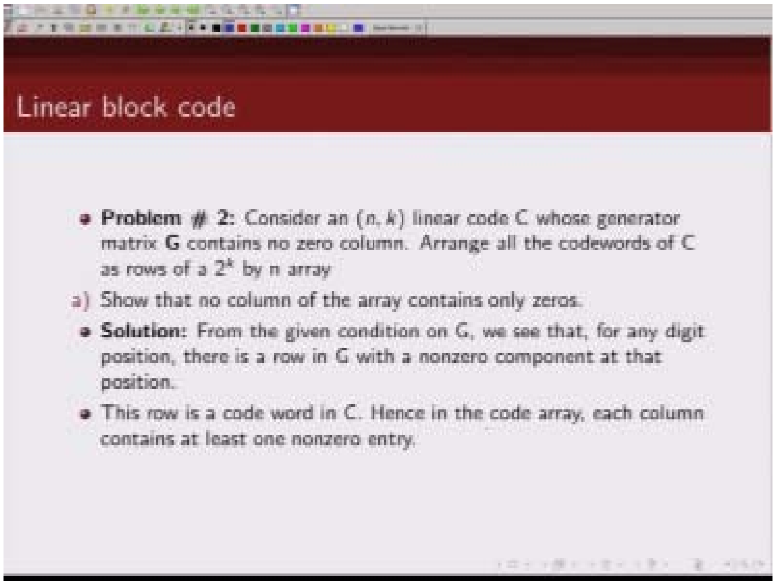
- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

a) Show that no column of the array contains only zeros.

- **Solution:** From the given condition on G, we see that, for any digit position, there is a row in G with a nonzero component at that position.
- This row is a code word in C. Hence in the code array, each column contains at least one nonzero entry.

And if this is true, what are the rows of, how do we generate the code words? We generate the code words by linear combination of these rows of this generator matrix.

(Refer Slide Time: 08:35)



Linear block code

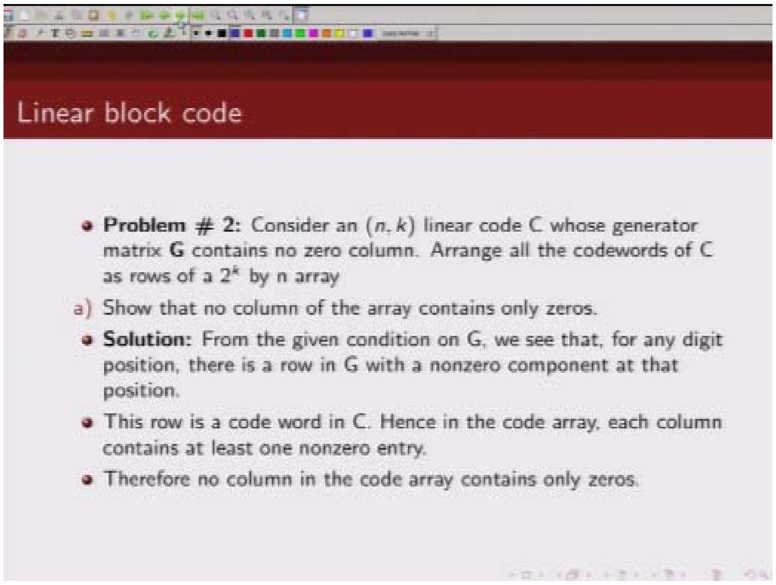
- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

a) Show that no column of the array contains only zeros.

- **Solution:** From the given condition on G, we see that, for any digit position, there is a row in G with a nonzero component at that position.
- This row is a code word in C. Hence in the code array, each column contains at least one nonzero entry.

And since the generator matrix does not contain any zero column, so each of these rows can be looked up as code word in C. So when we generate the code words using this generator matrix in this code array each column will have at least one non zero entry.

(Refer Slide Time: 08:57)



Linear block code

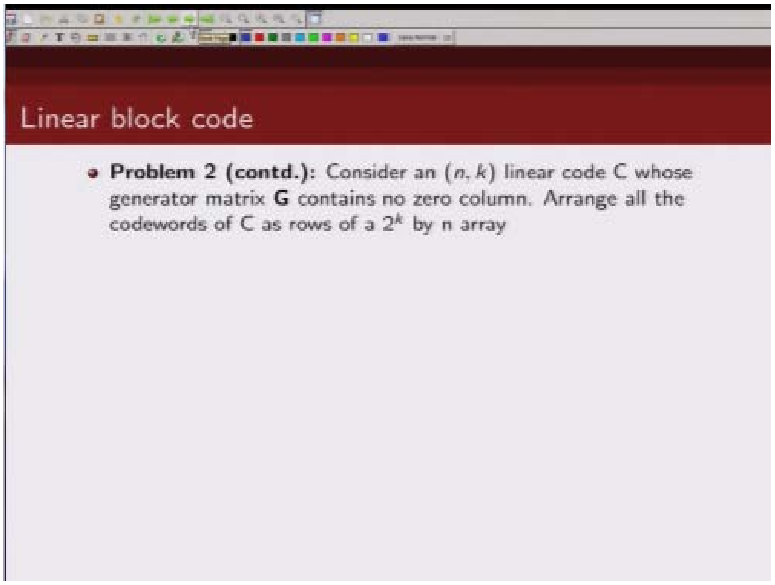
- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

a) Show that no column of the array contains only zeros.

- **Solution:** From the given condition on G, we see that, for any digit position, there is a row in G with a nonzero component at that position.
- This row is a code word in C. Hence in the code array, each column contains at least one nonzero entry.
- Therefore no column in the code array contains only zeros.

So this follows from the fact that our generator matrix G does not contain any zero column, and hence no column in this code array will have zeros.

(Refer Slide Time: 09:13)

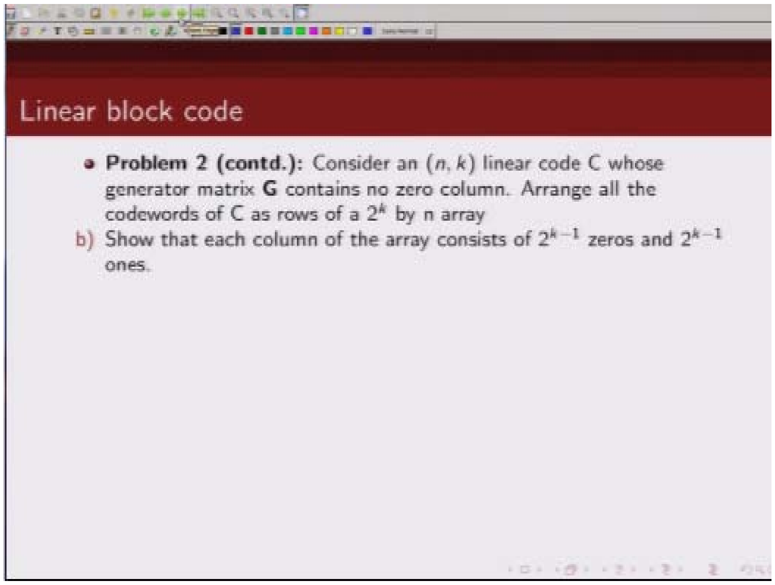


Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

The next result that we are

(Refer Slide Time: 09:15)



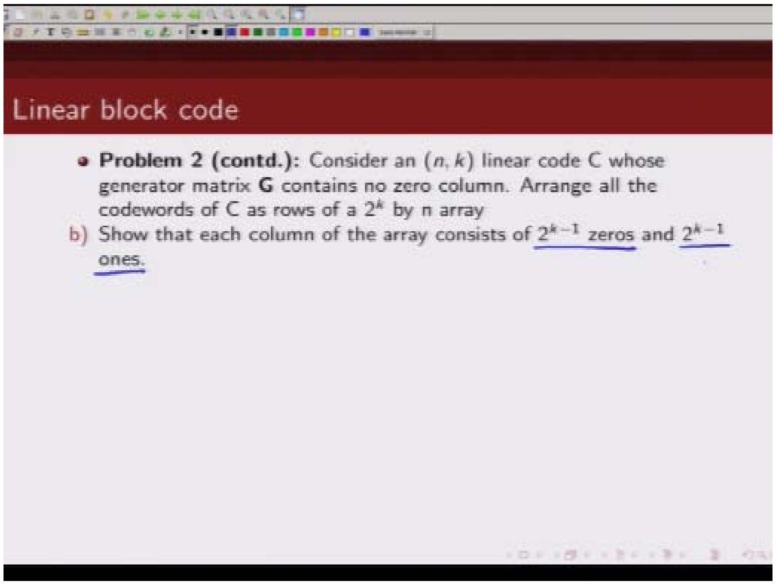
Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

Going to show is in this array in this 2^k x n array each column consists of.

(Refer Slide Time: 09:27)



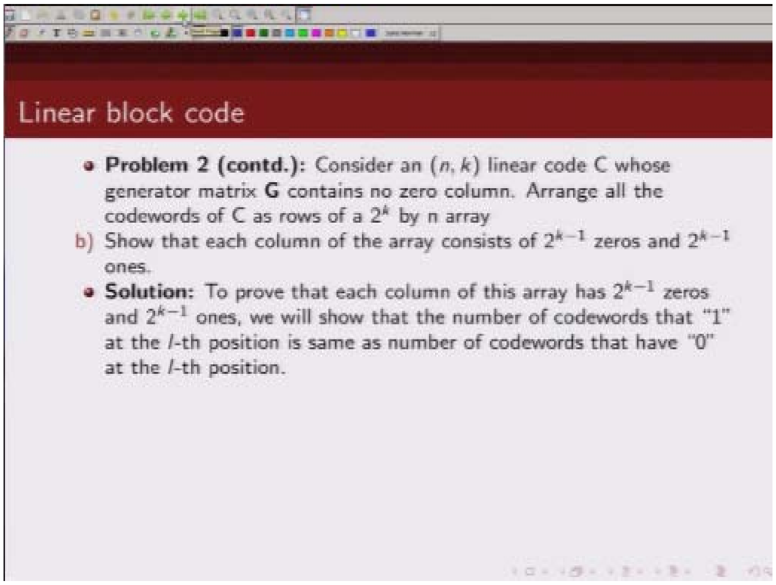
Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

Equal numbers of zeros and ones there are total 2^{k-1} zeros and2^{k-1} ones

(Refer Slide Time: 09:42)



Linear block code

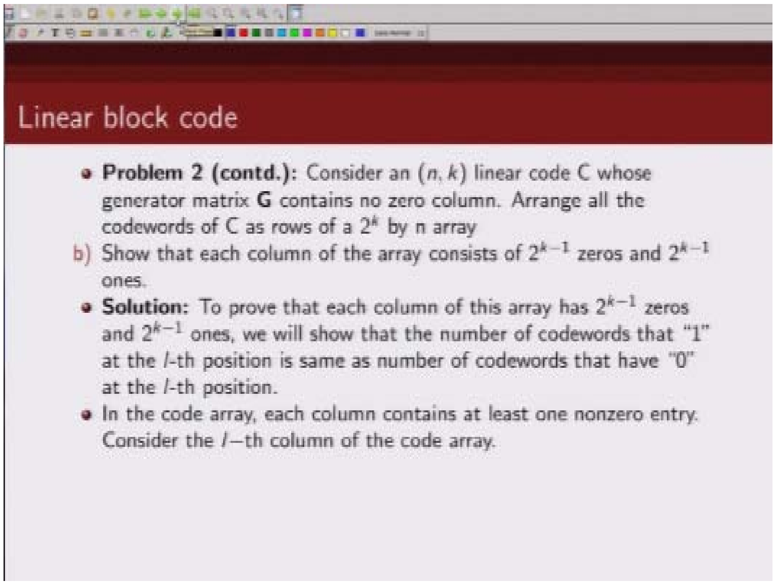
- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l-th position is same as number of codewords that have "0" at the l-th position.

So to prove this what we will do is we show that number of code of words that have 1 at l-th location is same as number of code words that have 0 at l-th location and in this way we will prove that this array has same number of zeros and ones.

(Refer Slide Time: 10:07)



Linear block code

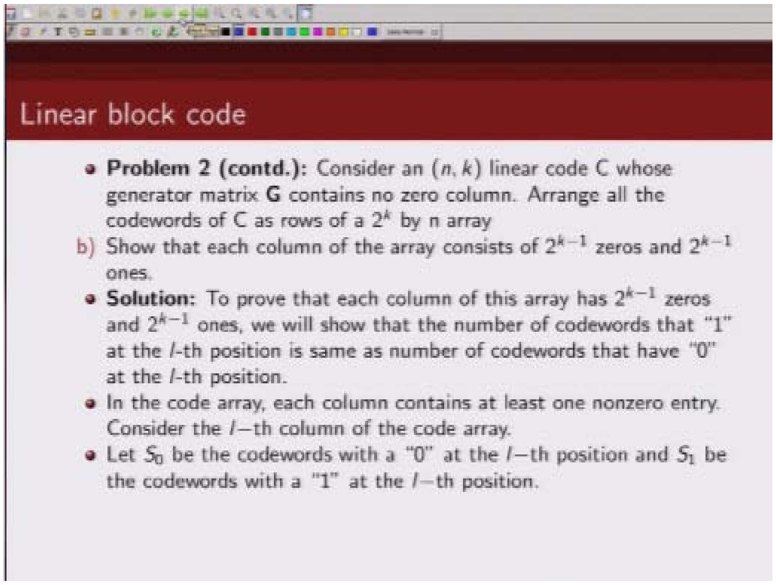
- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l-th position is same as number of codewords that have "0" at the l-th position.
- In the code array, each column contains at least one nonzero entry. Consider the $l-$th column of the code array.

So in this code array we know that each column will have at least one non zero entry that we proved in the earlier result so consider the l-th column of this code array.

(Refer Slide Time: 10:18)



Linear block code

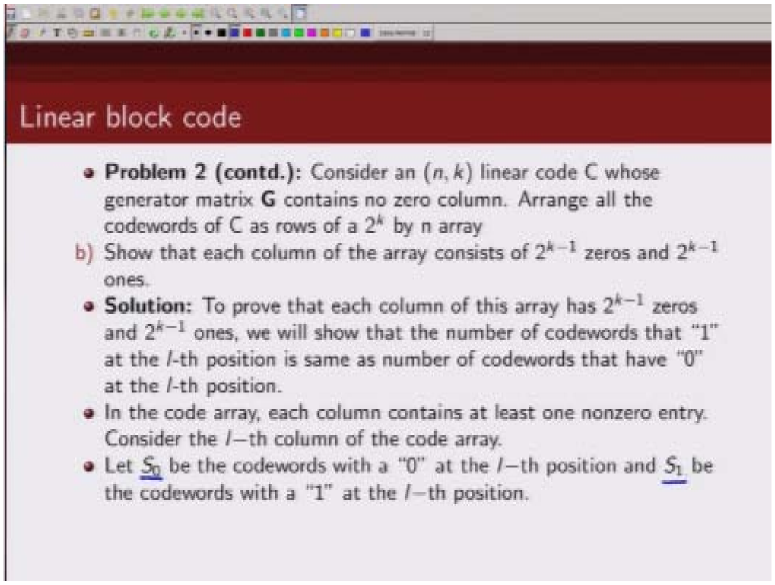
- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l-th position is same as number of codewords that have "0" at the l-th position.
- In the code array, each column contains at least one nonzero entry. Consider the $l-$th column of the code array.
- Let S_0 be the codewords with a "0" at the $l-$th position and S_1 be the codewords with a "1" at the $l-$th position.

Let us denote by

(Refer Slide Time: 10:22)



Linear block code

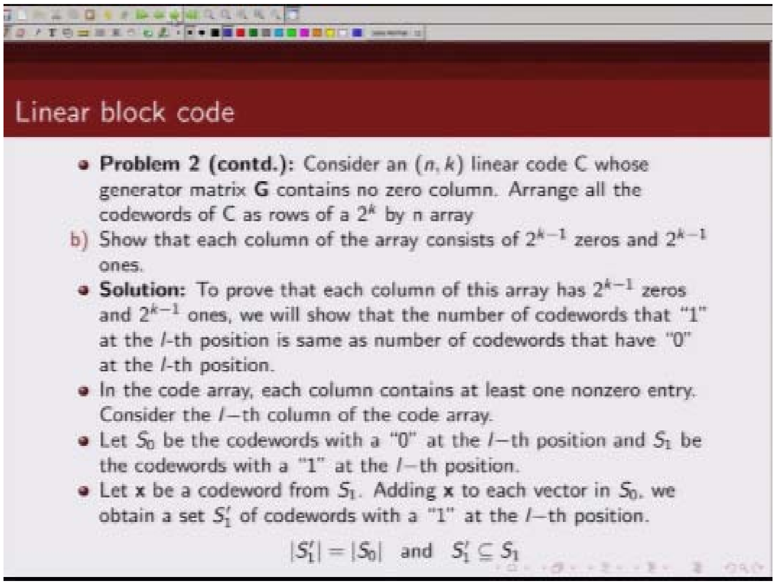
- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix $\mathbf{G}$ contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l-th position is same as number of codewords that have "0" at the l-th position.
- In the code array, each column contains at least one nonzero entry. Consider the $l-$th column of the code array.
- Let $\underline{S_0}$ be the codewords with a "0" at the $l-$th position and $\underline{S_1}$ be the codewords with a "1" at the $l-$th position.

S0 the set of code words that have 0 at the l-th location and let us denote by S1 the set of code words that have one at the l-th location

(Refer Slide Time: 10:41)



Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

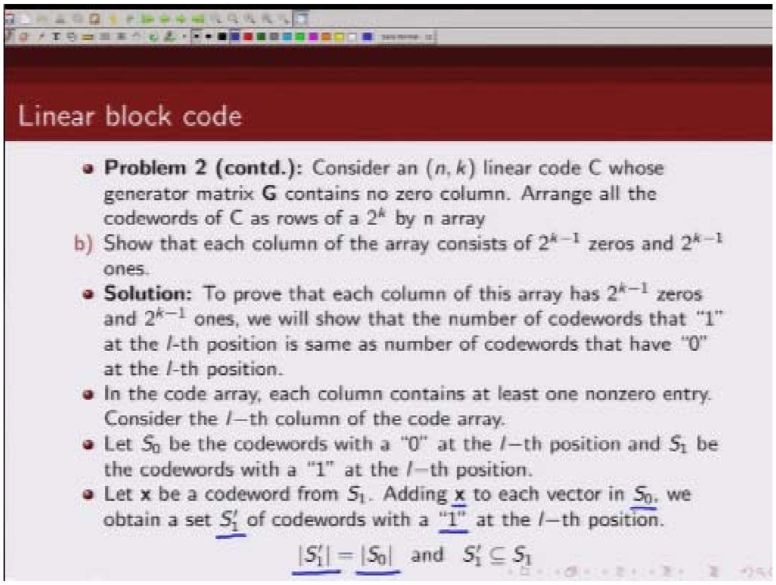
b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l-th position is same as number of codewords that have "0" at the l-th position.
- In the code array, each column contains at least one nonzero entry. Consider the $l-$th column of the code array.
- Let S_0 be the codewords with a "0" at the $l-$th position and S_1 be the codewords with a "1" at the $l-$th position.
- Let **x** be a codeword from S_1. Adding **x** to each vector in S_0, we obtain a set S_1' of codewords with a "1" at the $l-$th position.

$$|S_1'| = |S_0| \quad \text{and} \quad S_1' \subseteq S_1$$

Now we pick up an a code word x from the set S_1 that means x has 1 at l-th location now if we add this

(Refer Slide Time: 10:58)



Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix **G** contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array

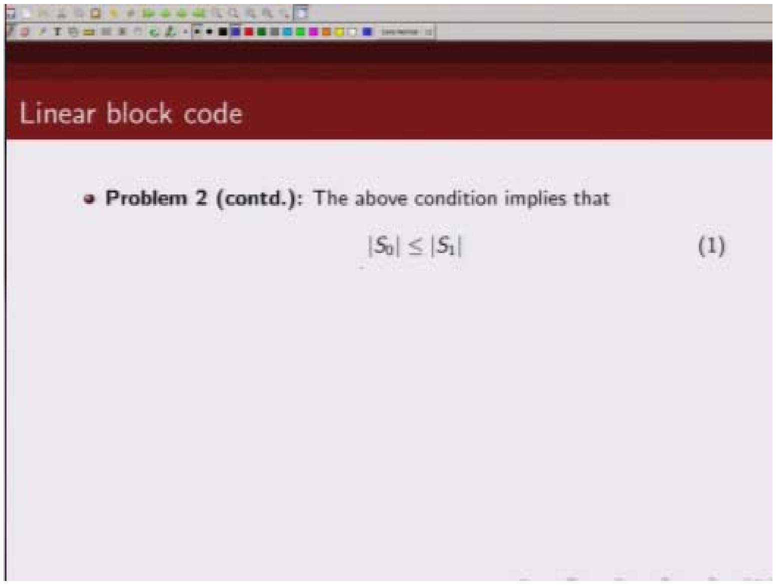
b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l-th position is same as number of codewords that have "0" at the l-th position.
- In the code array, each column contains at least one nonzero entry. Consider the $l-$th column of the code array.
- Let S_0 be the codewords with a "0" at the $l-$th position and S_1 be the codewords with a "1" at the $l-$th position.
- Let **x** be a codeword from S_1. Adding $\underline{\mathbf{x}}$ to each vector in $\underline{S_0}$, we obtain a set $\underline{S_1'}$ of codewords with a $\underline{\text{"1"}}$ at the $l-$th position.

$$\underline{|S_1'|} = \underline{|S_0|} \quad \text{and} \quad S_1' \subseteq S_1$$

Code word x to all the elements in the set S_0 what do we get, what we will get is a set containing 1 at l-th location, why because S_0 is a set that has 0 at the l-th location and x has x is taken from the set S_1 so x has 1 at l-th location, so if we add x to S elements in S_0 what we will get is there will be a I at the l-th bit location, so we denote this class of code word by S_1' and this S_1' will have 1 at the l-th location and since this S_1' is generated by adding x to this set of vectors in S_0 so number of elements in S_0 is going to be same as number of elements in S_1' and S_1' is a sub set of S_1 which is the set of all code words which has 1 at l-th location.

(Refer Slide Time: 12:20)



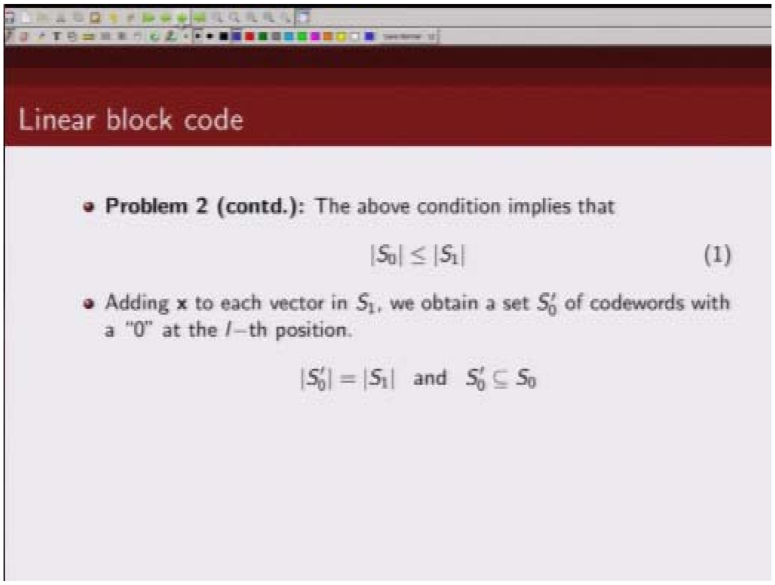
Linear block code

- **Problem 2 (contd.):** The above condition implies that

$$|S_0| \leq |S_1| \qquad (1)$$

So from this we get this condition that set of code words which has 0 at l-th location is less than equal to set of

(Refer Slide Time: 12:33)



Linear block code

- **Problem 2 (contd.):** The above condition implies that

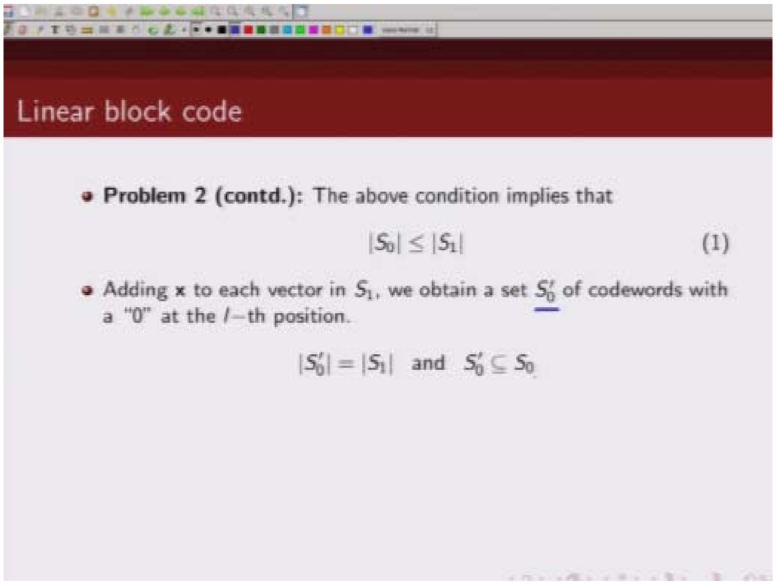
$$|S_0| \leq |S_1| \qquad (1)$$

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the $l-$th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

Code word which has 1 at l-th location, now add this same vector x which has 1 at l-th location to all the elements in S_1. When we do that what we get is a new set of vectors which has 0 at l-th location

(Refer Slide Time: 12:58)



Linear block code

- **Problem 2 (contd.):** The above condition implies that

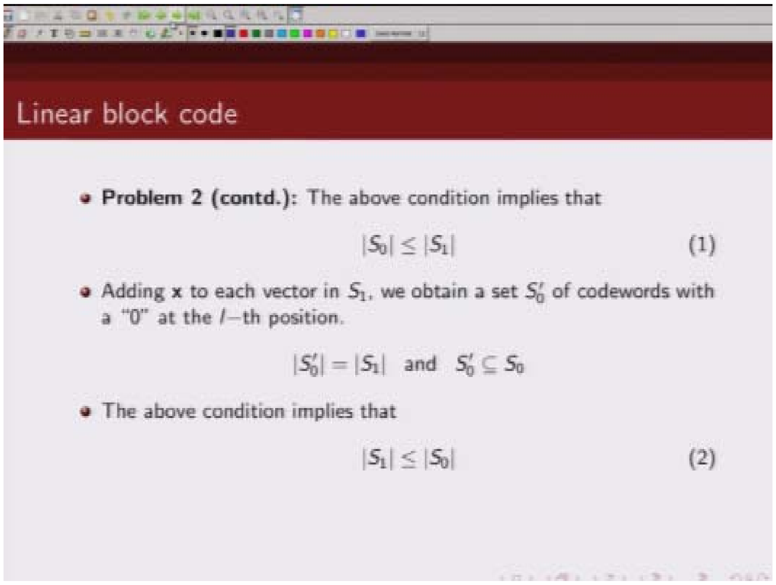
$$|S_0| \leq |S_1| \qquad (1)$$

- Adding **x** to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

We denote this set by S0 ´ so S0 ´ is a set of code words which are obtained by adding x to the set of vectors set of odd vectors which have 1 at l-th location, so then we can write thus the set of vectors in S0 ´ is same as set vectors in S1 and since S0 ´ is a subset of S0 what we can write

(Refer Slide Time: 13:32)



Linear block code

- **Problem 2 (contd.):** The above condition implies that

$$|S_0| \leq |S_1| \qquad (1)$$

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

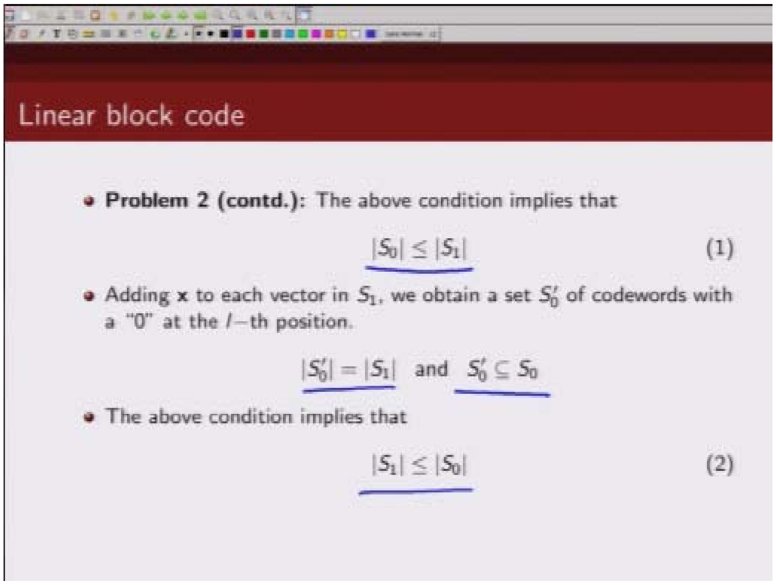
$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

- The above condition implies that

$$|S_1| \leq |S_0| \qquad (2)$$

Then is from this relation

(Refer Slide Time: 13:33)



Linear block code

- **Problem 2 (contd.):** The above condition implies that

$$\underline{|S_0| \leq |S_1|} \qquad (1)$$

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

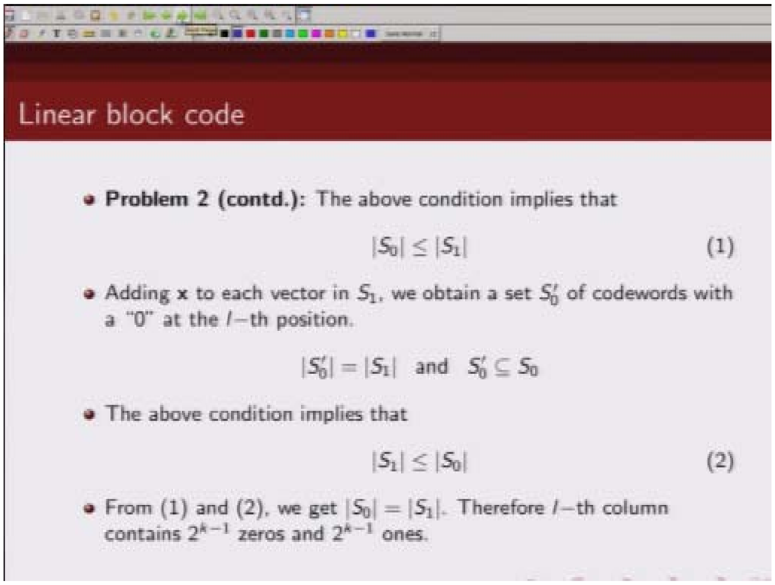
$$\underline{|S_0'| = |S_1|} \quad \text{and} \quad \underline{S_0' \subseteq S_0}$$

- The above condition implies that

$$\underline{|S_1| \leq |S_0|} \qquad (2)$$

And this relation we can write set of code words which have 1 at l-th location it is less than set of code words which has 0 at l-th location. Now equation 1 and 2 they are going to be simultaneously satisfied only when this is satisfied with equality, so then what it shows

(Refer Slide Time: 14:03)



Linear block code

- **Problem 2 (contd.):** The above condition implies that

$$|S_0| \leq |S_1| \qquad (1)$$

- Adding **x** to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the $l-$th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

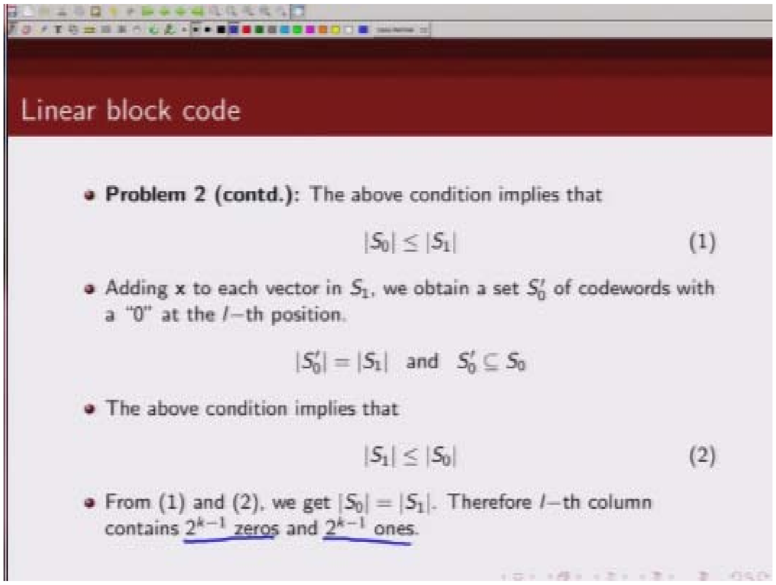
- The above condition implies that

$$|S_1| \leq |S_0| \qquad (2)$$

- From (1) and (2), we get $|S_0| = |S_1|$. Therefore $l-$th column contains 2^{k-1} zeros and 2^{k-1} ones.

Here is that at any l-th location number of code words which have 0 at l-th location is same as number of code words which have 1 at l-th location so basically each column will then have

(Refer Slide Time: 14:22)



Linear block code

- **Problem 2 (contd.):** The above condition implies that

$$|S_0| \leq |S_1| \qquad (1)$$

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the $l-$th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

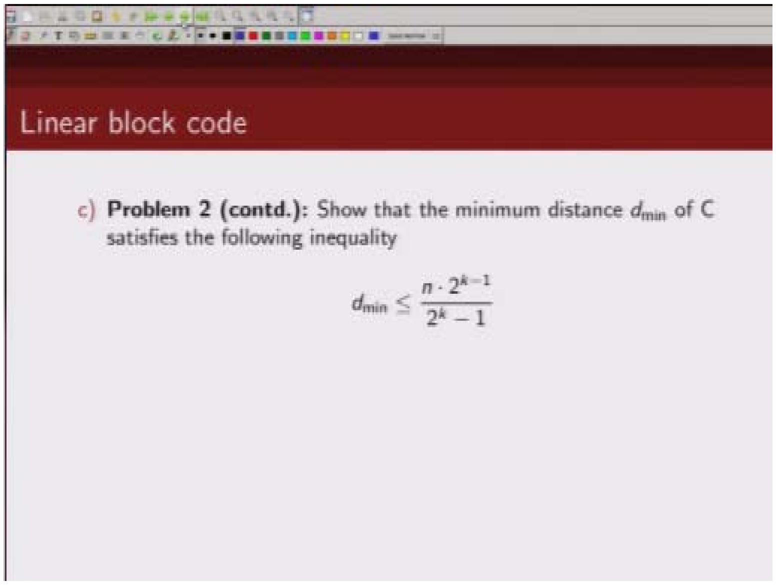
- The above condition implies that

$$|S_1| \leq |S_0| \qquad (2)$$

- From (1) and (2), we get $|S_0| = |S_1|$. Therefore $l-$th column contains $\underline{2^{k-1}\text{ zeros}}$ and $\underline{2^{k-1}\text{ ones}}$.

Same number of zeros and same number of ones.

(Refer Slide Time: 14:26)



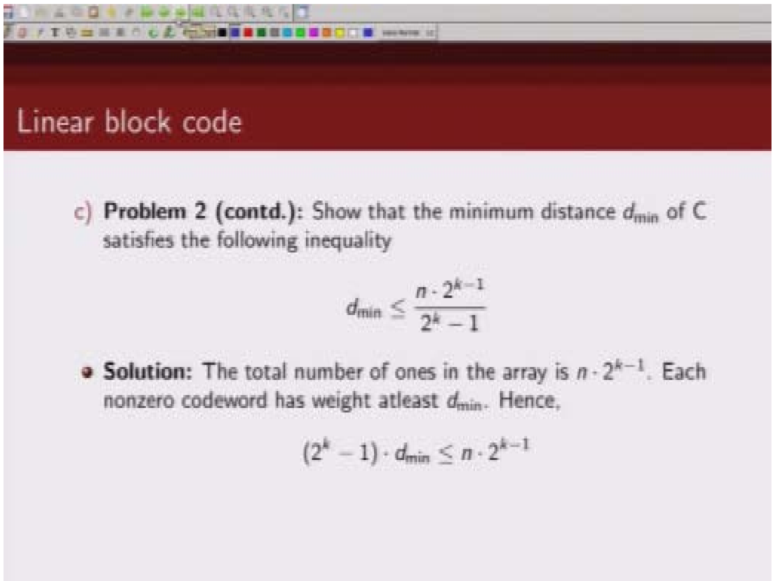
Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance $d_{\min}$ of C satisfies the following inequality

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

Now we prove another result, we showed that minimum distance of the code is upper bounded by this quantity and to prove this result we are just going to use the result we just proved in the previous section.

(Refer Slide Time: 14:47)



Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance $d_{\min}$ of C satisfies the following inequality

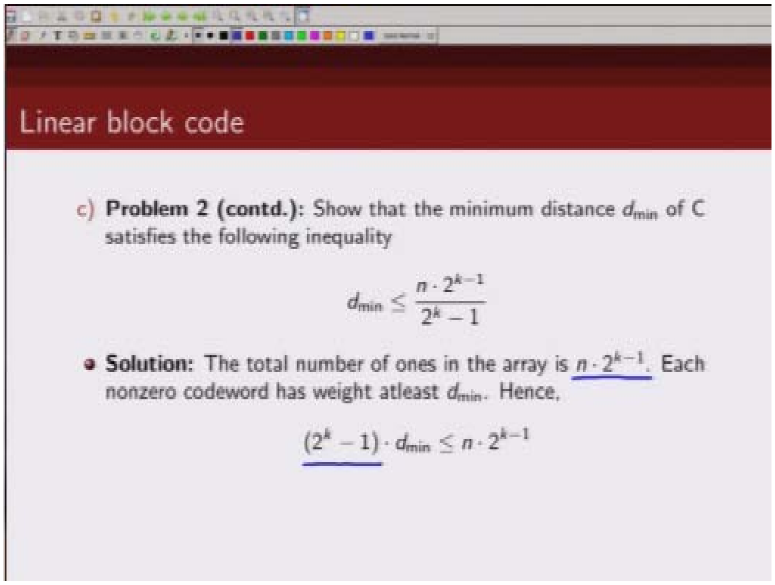
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

- **Solution:** The total number of ones in the array is $n \cdot 2^{k-1}$. Each nonzero codeword has weight atleast $d_{\min}$. Hence,

$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$

So in the previous section what we did was we arranged this 2^k code words in an array, 2^k x n array and we showed that each column of this array has 2^{k-1} ones and 2^{k-1} zeros, so in this whole array which has n columns total number of ones.

(Refer Slide Time: 15:160



Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance d_{min} of C satisfies the following inequality

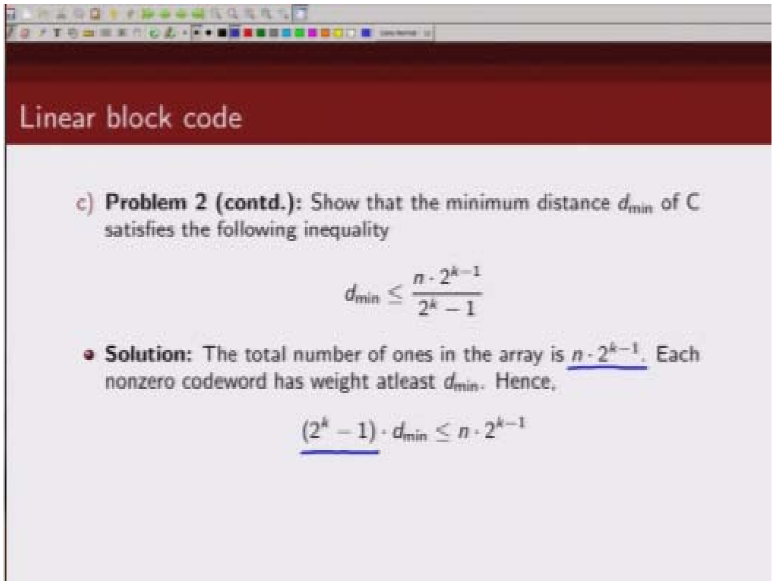
$$d_{min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

- **Solution:** The total number of ones in the array is $\underline{n \cdot 2^{k-1}}$. Each nonzero codeword has weight atleast d_{min}. Hence,

$$\underline{(2^k - 1)} \cdot d_{min} \leq n \cdot 2^{k-1}$$

Is given by this, n times $2^{k-1,}$ now since each non zero code word will have minimum distance at least d_{min} and how many total code words we have, 2^k one of them is all zero code word so how many non zero code words we have, that is given by 2^k-1 and each of these non zero code words have minimum distance at least $d_{min,}$ so total number of non zero code word multiplied by d_{min} must be less the equal to

(Refer Slide Time: 16:01)



Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance $d_{\min}$ of C satisfies the following inequality

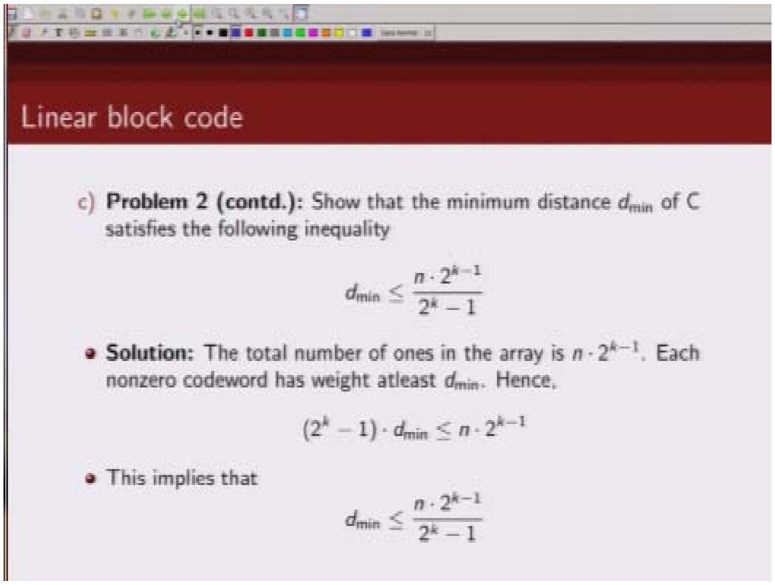
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

- **Solution:** The total number of ones in the array is $\underline{n \cdot 2^{k-1}}$. Each nonzero codeword has weight atleast $d_{\min}$. Hence,

$$\underline{(2^k - 1)} \cdot d_{\min} \leq n \cdot 2^{k-1}$$

Total number of ones in this code array which is given by n times 2^k-1 2^{k-1}, so from this relation then we can then

(Refer Slide Time: 16:15)



Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance $d_{\min}$ of C satisfies the following inequality

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

- **Solution:** The total number of ones in the array is $n \cdot 2^{k-1}$. Each nonzero codeword has weight atleast $d_{\min}$. Hence,

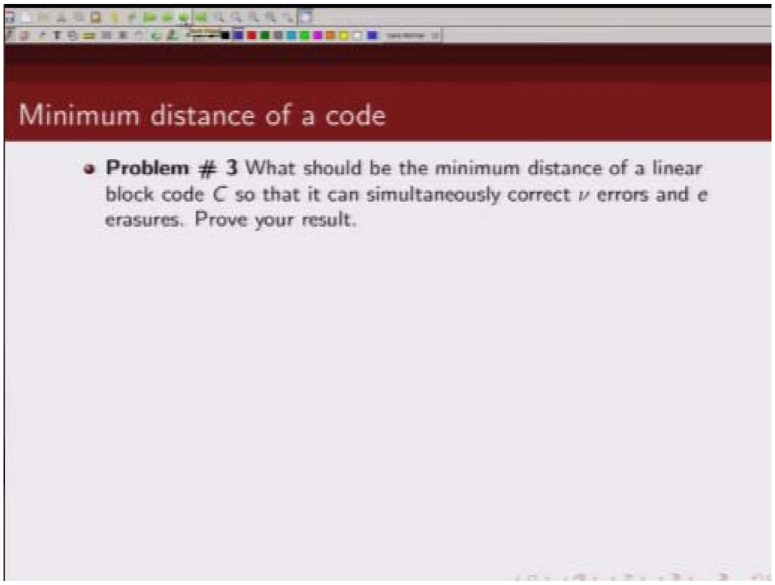
$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$

- This implies that

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

Write that minimum distance of a code is upper bounded by this relationship.

(Refer Slide Time: 16:23)

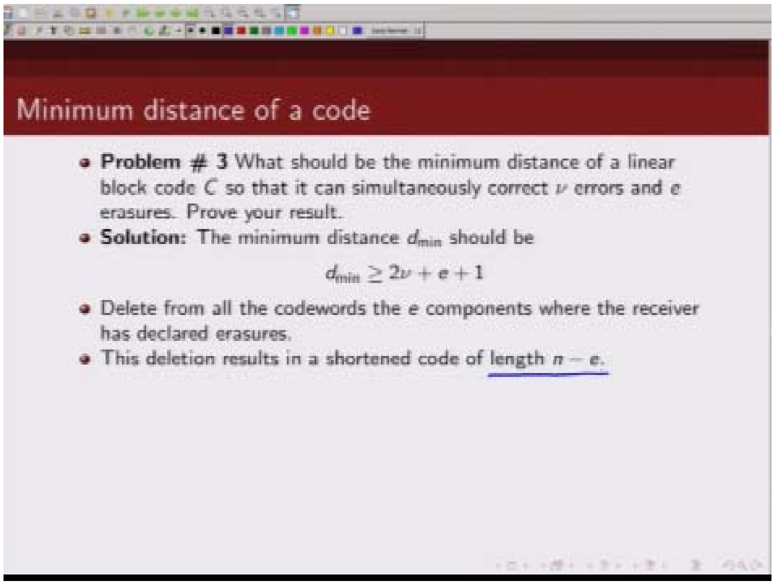


Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.

So next problem that we will look at is what is a minimum distance of a linear block code C that can simultaneously correct μ errors and e erasures? Now just recall what do we mean by error correction and error erasure correction. Basically so erasure is basically some of the bits are getting erased, so you send n bits if e bits are getting erased what you are receiving is n-e bits, and error correction you are familiar with basically we want to correct errors that have happened in so many bit locations.

(Refer Slide Time: 17:08)



Minimum distance of a code

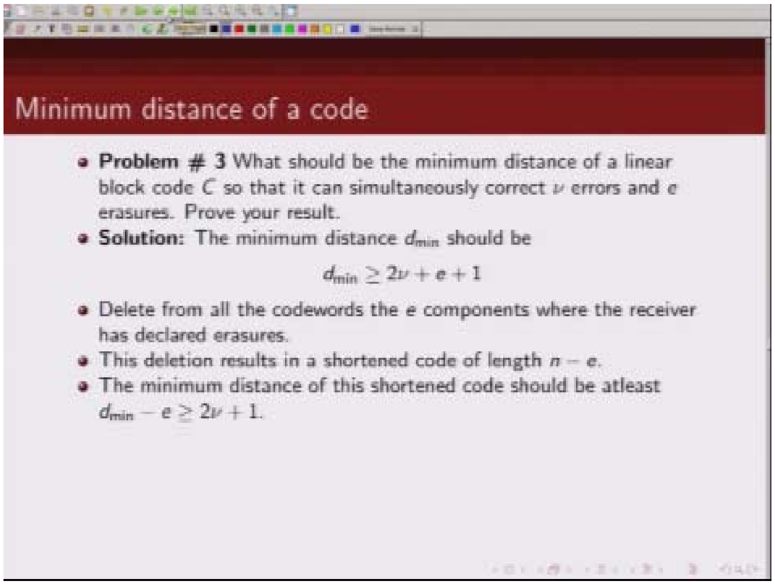
- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance $d_{\min}$ should be

$$d_{\min} \geq 2\nu + e + 1$$

- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.

So the question is what should be the minimum distance of a linear block code that can simultaneously correct μ errors as wells as e- erasures? Now if the minimum distance of a code is at least 2μ + e + 1 then it can simultaneously correct μ errors and e-erasures, we are going to next prove this result. So delete from all code words e components which got erased, if we delete these e components what we are left is n – e length shortened code word, so this deletion of e-component results in a shortened code of length n – e.

(Refer Slide Time: 18:05)



Minimum distance of a code

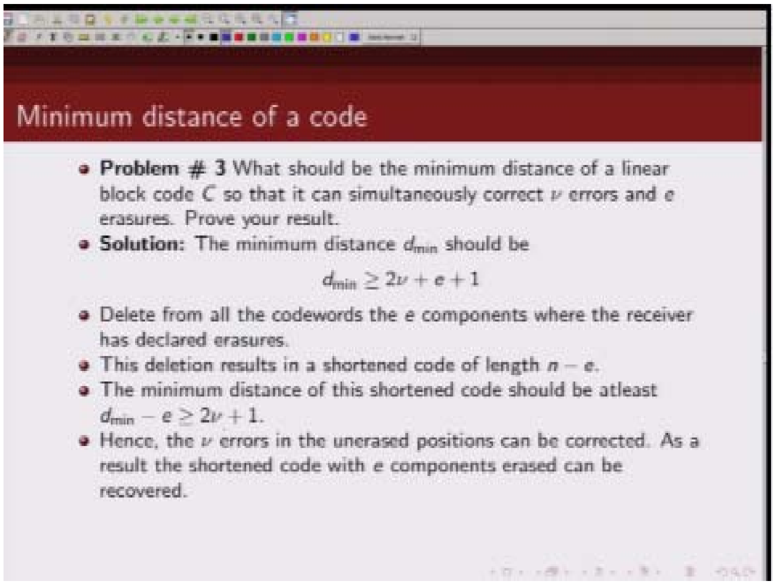
- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance $d_{\min}$ should be

$$d_{\min} \geq 2\nu + e + 1$$

- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.

Now we know that if we want to correct t errors what should be the minimum distance of the code, it should be at least 2t + 1 so this code.

(Refer Slide Time: 18:18)



Minimum distance of a code

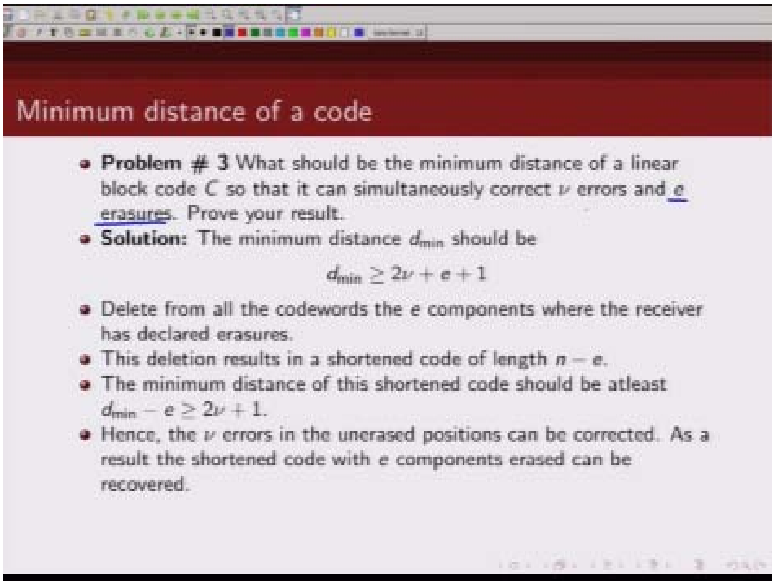
- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance $d_{\min}$ should be

$$d_{\min} \geq 2\nu + e + 1$$

- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.

Basically if we want to correct µ errors the minimum distance of the code after this e-erasures should be greater than equal to 2µ + 1, so if minimum distance of the code after this e-erasures the minimum distance is still larger than 2µ + 1 then this code can correct µ errors, so we want our minimum distance of the code to be at least 2µ + e + 1. Now since this µ errors are in the un-erased positions can be corrected it this condition holds, so as a result basically we would be able to correct µ errors, now remember we have to simultaneously.

(Refer Slide Time: 19:17)



Minimum distance of a code

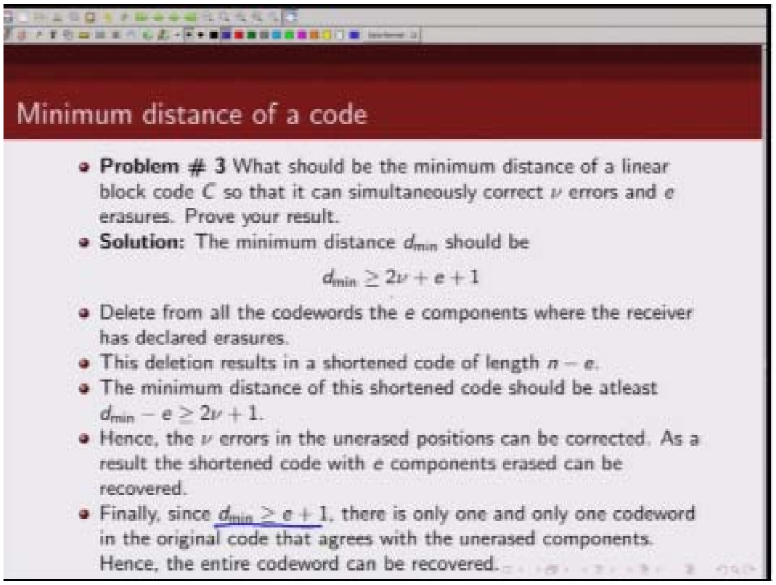
- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance $d_{\min}$ should be

$$d_{\min} \geq 2\nu + e + 1$$

- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.

Also be able to basically it would be not only simultaneously correct μ errors but we have to correct e-erasures also. Now what is the condition on minimum distance such that e-erasures can be also corrected? The minimum distance of the code should be at least greater than number of erasures + 1, so if the minimum distance of the code.

(Refer Slide Time: 19:46)



Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance $d_{\min}$ should be

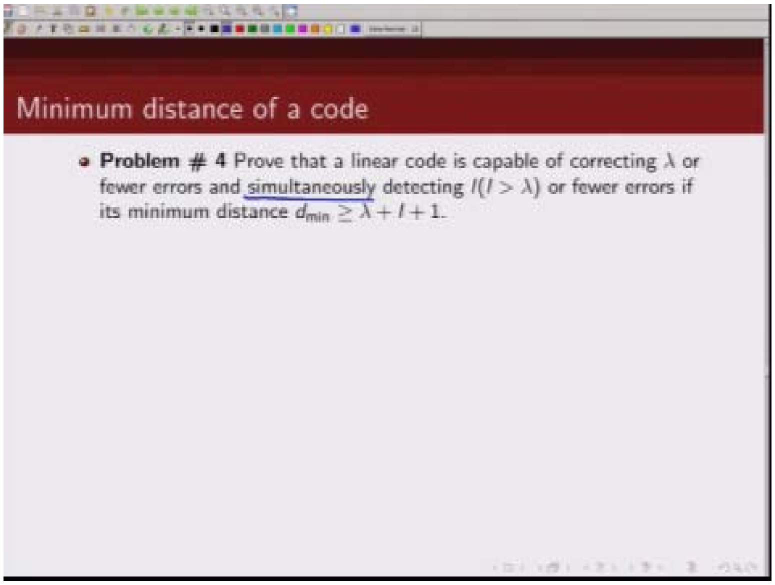
$$d_{\min} \geq 2\nu + e + 1$$

- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.
- Finally, since $\underline{d_{\min} \geq e + 1}$, there is only one and only one codeword in the original code that agrees with the unerased components. Hence, the entire codeword can be recovered.

Is greater than e + 1 then there is only 1 code word in the original code that maps to the shortened codes, so as long as minimum distance of the code is greater than e + 1 there is only one 1 code word, the original code that agrees with the un-erased component so this as long as minimum distance of the code is greater than e + 1 there is only 1 code that maps from erased shortened code to the original code, and since in this case the d minimum is already 2µ + e + 1 which is greater than e + 1.

This code would be able to correct e-erasures as well, so if we choose our minimum distance of the code to be greater than equal to 2µ + e+ 1 it would be able to correct µ errors as well as e-erasures.

(Refer Slide Time: 20:59)

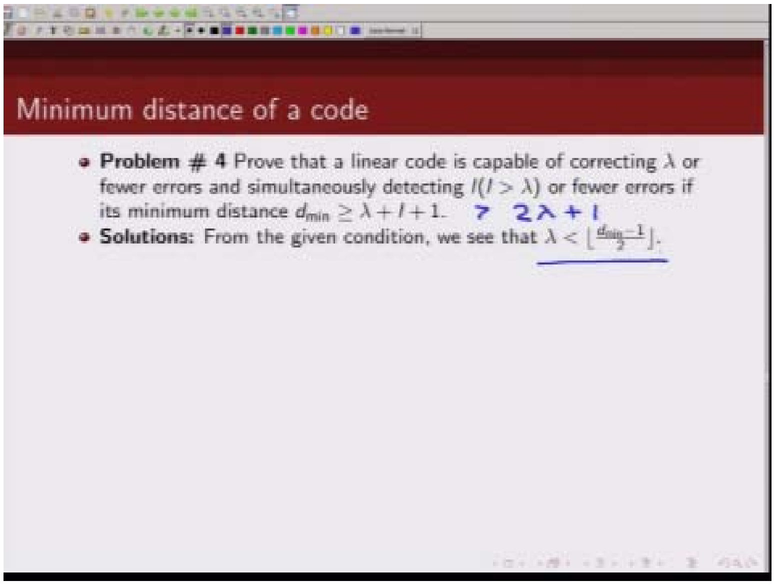


Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.

The next problem that we are going to solve is as follows. Prove that linear block code is capable of correcting λ pr fewer errors and simultaneously detecting L where L is greater than λ or fewer errors if the minimum distance of the code is at least λ + l + 1. Please pay attention to the word simultaneously, so we want not only to correct μ errors along with that we should be able to detect l errors as well, that is what we mean by simultaneous error.

(Refer Slide Time: 21:47)

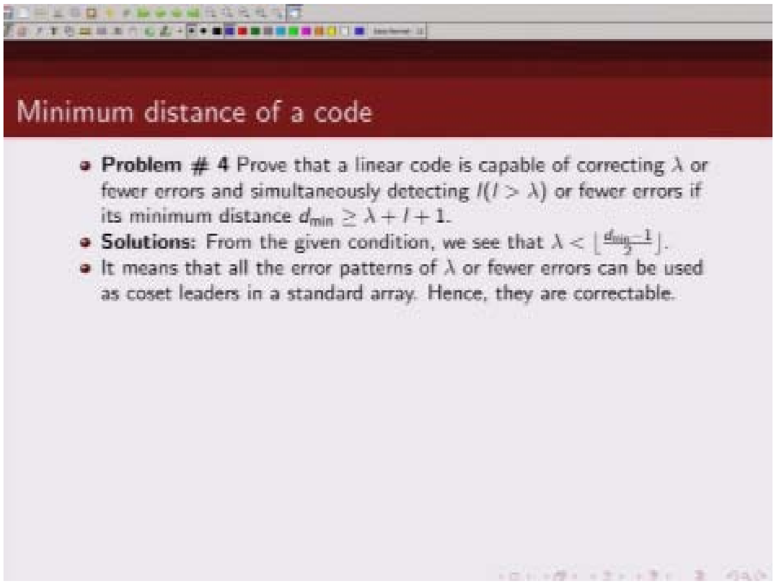


Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$. $> 2\lambda + 1$
- **Solutions:** From the given condition, we see that $\underline{\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor}$.

Detection and correction, so let us prove this result. Now note λ is less than l so if minimum distance is λ + l + 1 this is basically greater than 2 λ + 1, and if the minimum distance is greater than 2 λ + 1 it would be able to correct λ , so from this given condition that d_{min} is at least λ + 1 + 1 where l is greater than λ we know that the minimum distance is greater than 2 λ + 1 so it should be able to correct λ errors.

(Refer Slide Time: 22:41)

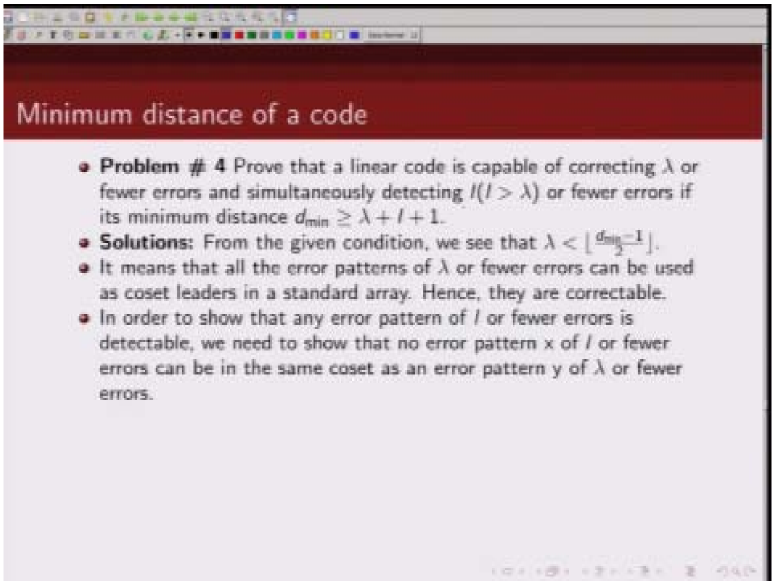


Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.

Now note we want to in addition to correcting λ or fewer errors we also want to simultaneously detect l errors. Now if we want to simultaneously detect those l or fewer errors we have to ensure thus those error patterns of weight l or less are not in the same coset as the error patterns.

(Refer Slide Time: 23:09)

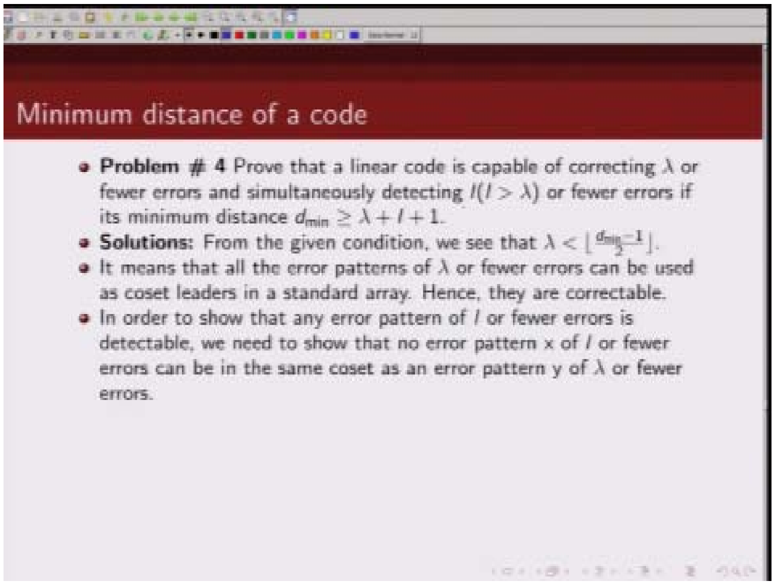


Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.

That we are trying to correct, now since λ errors can be corrected we can put all error patterns of λ or fewer errors as coset leader in our standard array and they can be correctable. Next to simultaneously detect l errors we have to show that none of these error patterns of weight l or less are in the same coset as these.

(Refer Slide Time: 23:44)

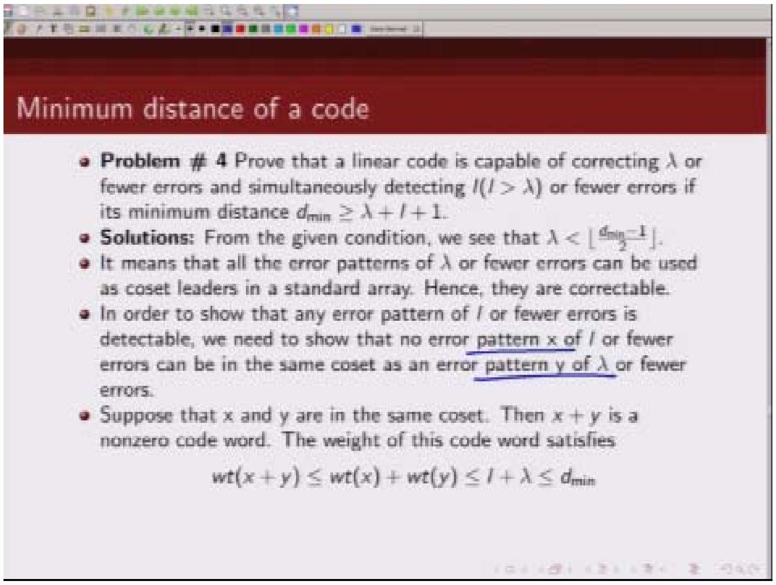


Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.

Error patterns of λ or less error, so we need to show that no error pattern x of length l or fewer errors are in the same coset as error pattern Y of λ or fewer errors. If they are in the same coset because we are using those coset leaders for error correction we would not be able to detect those error patterns, so it is important that those error patterns of weight l or less if we want to detect them they should not be in the same coset as the correctable.

(Refer Slide Time: 24:23)



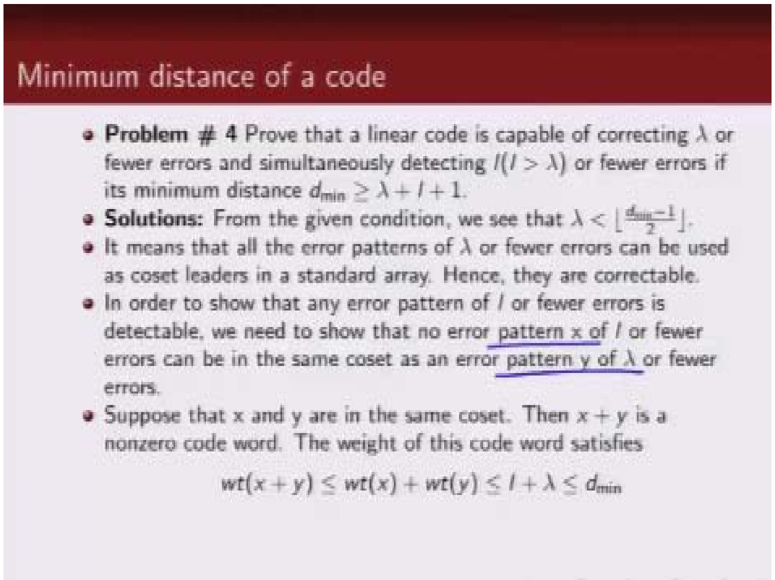
Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error $\underline{\text{pattern } x \text{ of}}$ l or fewer errors can be in the same coset as an error $\underline{\text{pattern } y \text{ of } \lambda}$ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{min}$$

Error patterns, so we are now going to use method of contradiction to show that it is not possible to have these error pattern x of l of fewer errors in the same coset as these error patterns y of λ of fewer errors which we are trying to correct. So how does this the method of contradiction work? We will first assume that they are in the same coset and then we will show that this is not possible, hence our assumption that they are in the same coset is wrong.

(Refer Slide Time: 25:01)



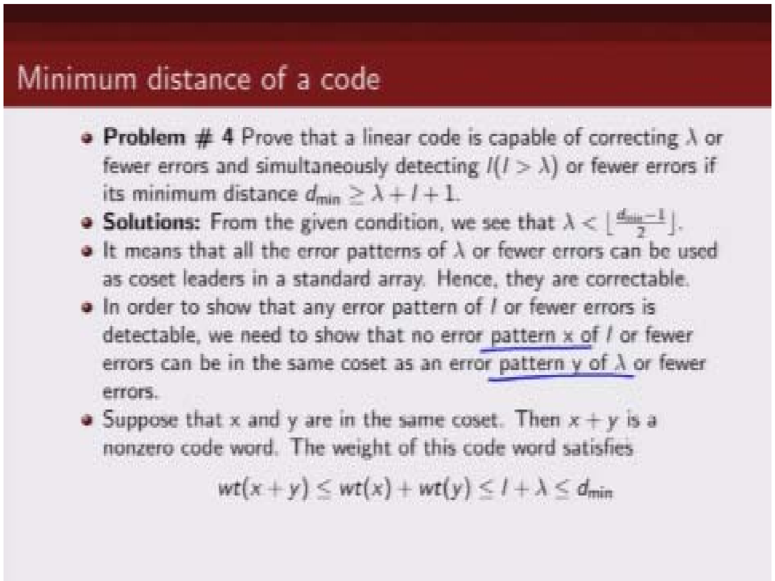
Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

So we start our proof by saying these error pattern x of weight l or less and error pattern y of weight λ or less they are in the same coset, now if x and y are in the same coset we know from our standard array.

(Refer Slide Time: 25:25)



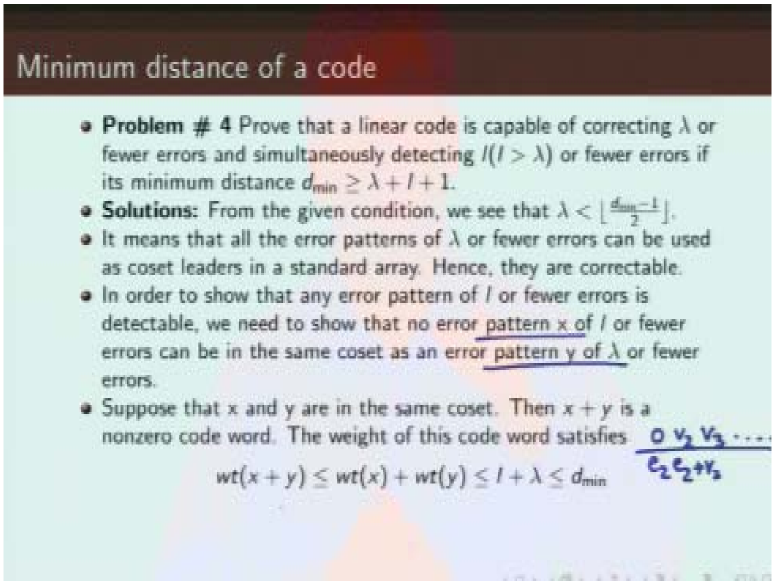
Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x+y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{min}$$

That x + y should be a non – zero code word, if you recall the entries our standard array we had in the first column an all zero code word and then we had other code word.

(Refer Slide Time: 25:33)



Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

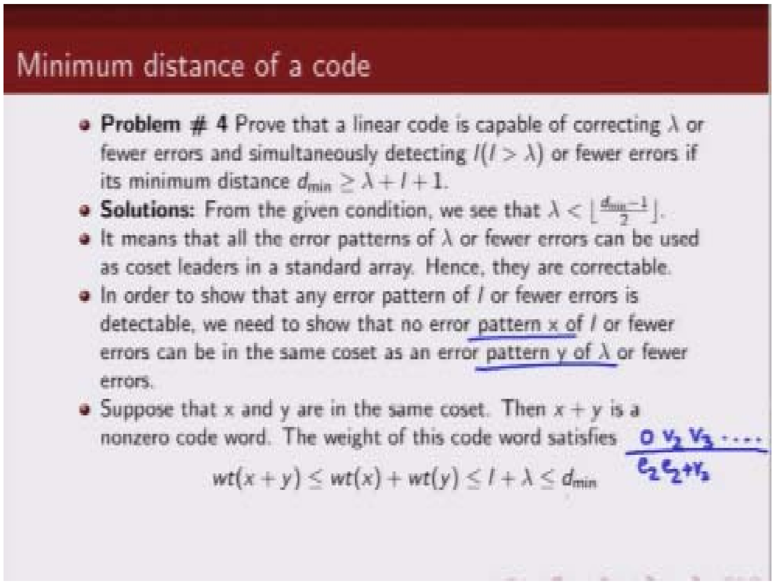
$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

$0 \; v_2 \; v_3 \cdots$

$e_2 \; e_2+v_2$

v2, v3… and then what we had was error pattern e2 and then we had basically this was e2+v2 like that we had and if you add any two elements of a coset or a row what you will notice is sum of them is a valid code word, so if x and y are in the same coset x + y must be a non – zero code word. Now let us look at what is the weight of x + y? So wt(x + y) is less than equal to wt(x) + wt(y) because it is possible that there are some common elements between x and y.

(Refer Slide Time: 26:33)



Minimum distance of a code

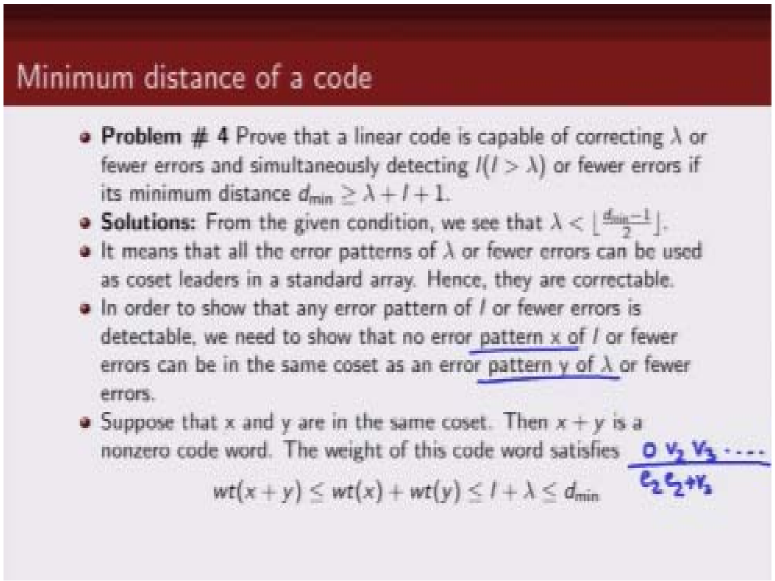
- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies $\frac{0 \; v_2 \; v_3 \cdots}{e_2 \; e_2+v_2}$

$$wt(x+y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

That is why the wt(x + y) is less than equal to wt(x) + wt(y) and what is weight of x? x are the error pattern of weight l or less, so the maximum weight of x is l, similarly maximum weight of y is λ, so weight of x + y is then less than equal to λ + l and what is the minimum distance? Minimum distance of code is atleast λ + l + 1, so weight of x + y is then less than d_{min}.

So what we have shown is, the weight of x + y, x + y should have been a code word is a code word if they are in the same coset, if x and y are in the same coset x + y is a valid non – zero code word, but what we have shown here is weight of x + y is less than d_{min}. So if x + y is a valid code word, its minimum weight should be atleast d_{min}.

(Refer Slide Time: 27:48)



Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

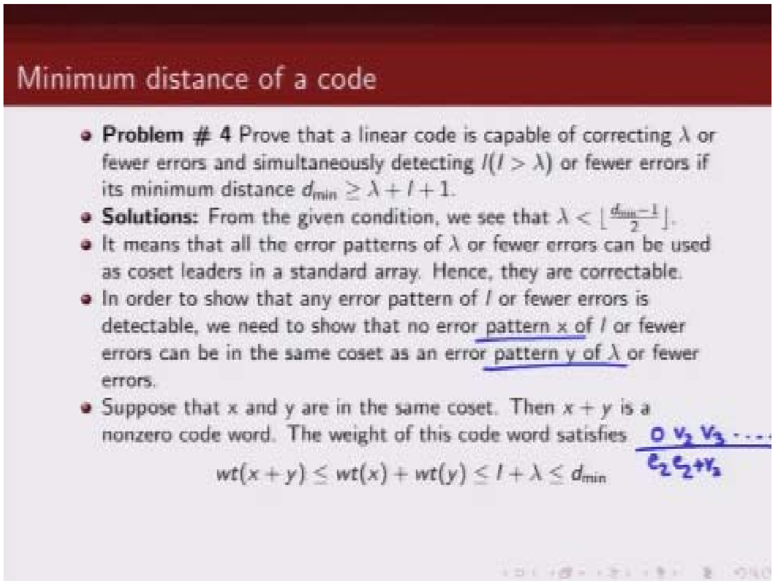
$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{min}$$

$0 \; v_2 \; v_3 \; \cdots$

$e_2 \; e_2 + v_2$

So from here basically what we get is, it is not possible to have x + y in the same coset because if they were in the same coset x + y would have been a valid code word and its weight of x + y should have been more than d_{min}, but here in this case it is coming out to be less than d_{min} , hence our assumption that x and y are in the same coset is wrong.

(Refer Slide Time: 28:23)



Minimum distance of a code

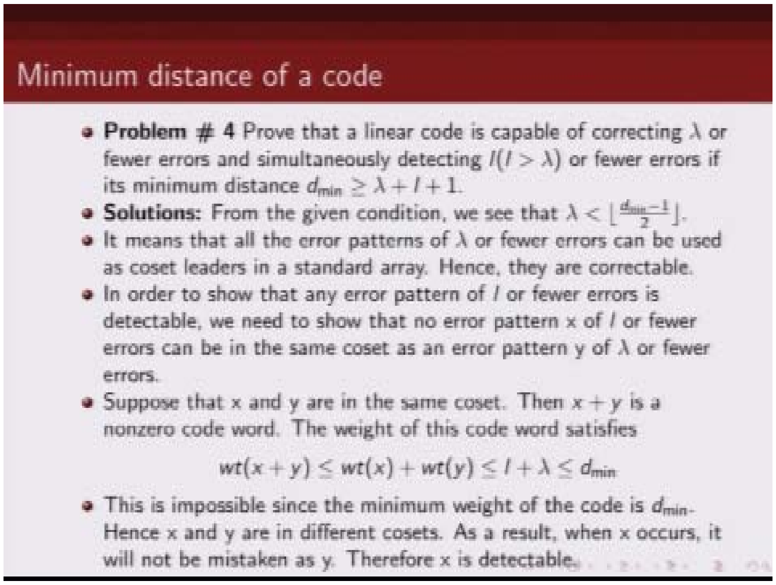
- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

$0 \; v_2 \; v_3 \; \cdots$

$e_2 \; e_2 + v_2$

(Refer Slide Time: 28:25)



Minimum distance of a code

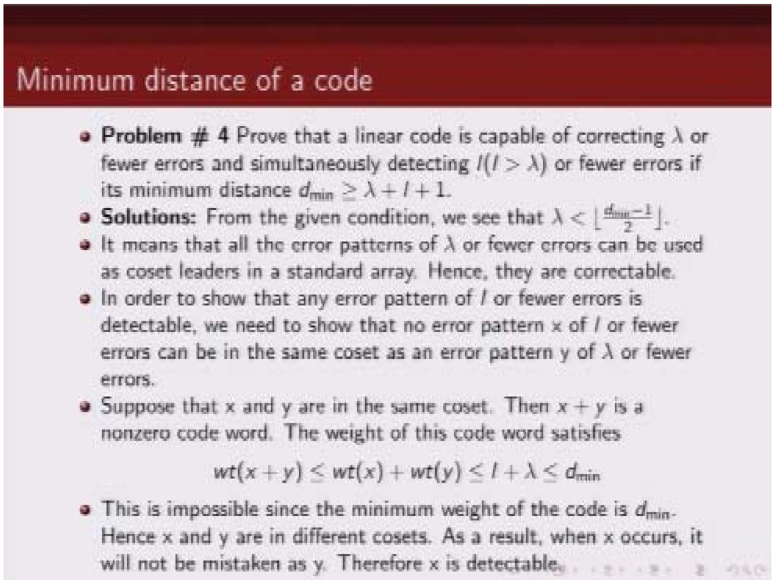
- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

- This is impossible since the minimum weight of the code is $d_{\min}$. Hence x and y are in different cosets. As a result, when x occurs, it will not be mistaken as y. Therefore x is detectable.

Now if x and y are not in the same coset then we can always put those error patterns of y and x in different cosets and hence we can simultaneously.

(Refer Slide Time: 28:39)



Minimum distance of a code

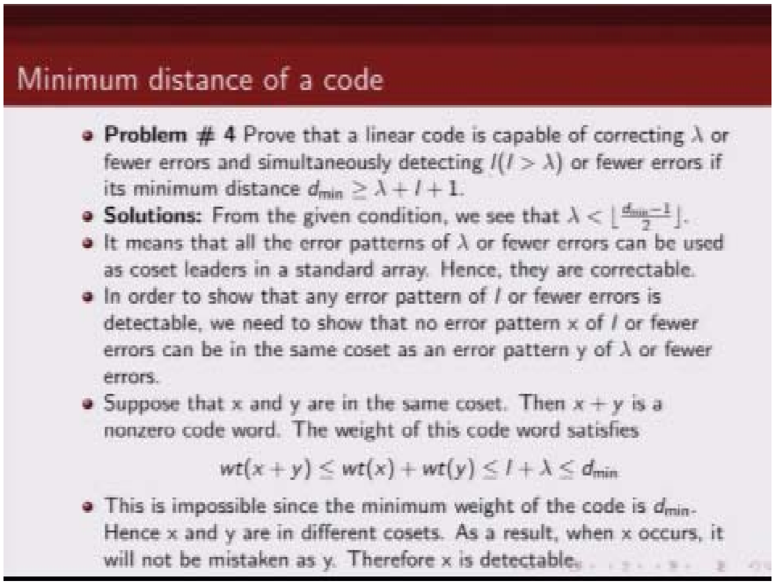
- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x+y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

- This is impossible since the minimum weight of the code is $d_{\min}$. Hence x and y are in different cosets. As a result, when x occurs, it will not be mistaken as y. Therefore x is detectable.

Detect and correct errors, so we can simultaneously correct λ errors while detecting also l errors, okay. So again to recap basically we prove this result by showing that if we want to simultaneously correct and detect errors those error patterns should be in the different cosets.

(Refer Slide Time: 29:11)



Minimum distance of a code

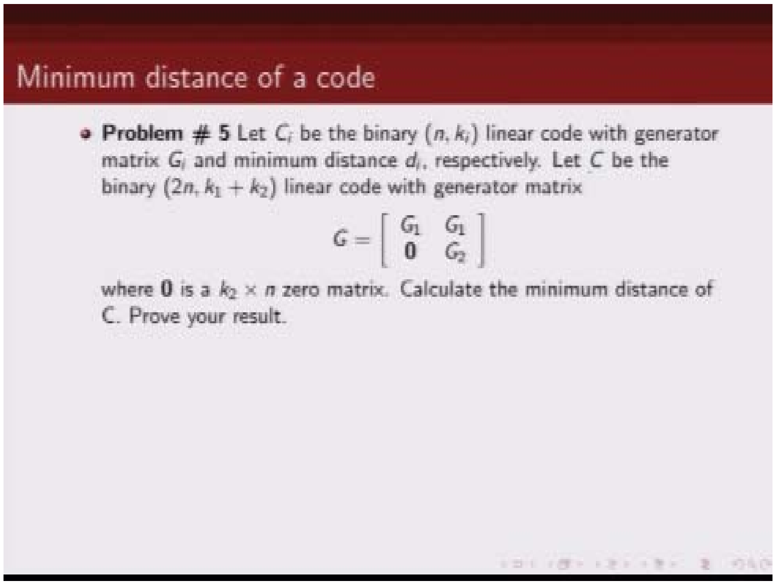
- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x+y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

- This is impossible since the minimum weight of the code is $d_{\min}$. Hence x and y are in different cosets. As a result, when x occurs, it will not be mistaken as y. Therefore x is detectable.

And hence we can simultaneously correct and detect those error patterns.

(Refer Slide Time: 29:16)



Minimum distance of a code

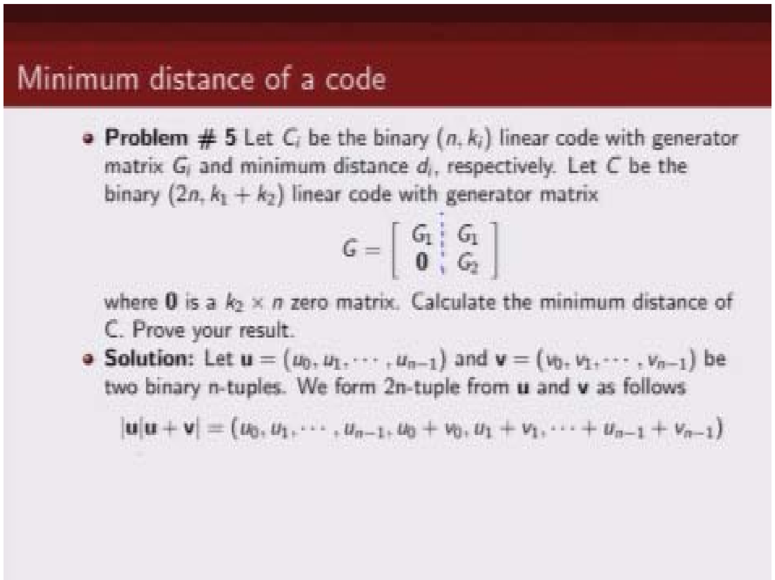
- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i, respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \begin{bmatrix} G_1 & G_1 \\ \mathbf{0} & G_2 \end{bmatrix}$$

where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C. Prove your result.

The next problem that we are going to solve is as follows. Let C_i be a binary linear code with code parameters given by (n, k_i) with generator matrix G_i and minimum distance d_i and let us consider a new code C, a new binary code linear code of length 2n and message bit length k_1+k_2 whose generator matrix is given by this expression. Now what is the minimum distance of this new code C?

(Refer Slide Time: 30:04)



Minimum distance of a code

- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i, respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \left[\begin{array}{c|c} G_1 & G_1 \\ \mathbf{0} & G_2 \end{array}\right]$$

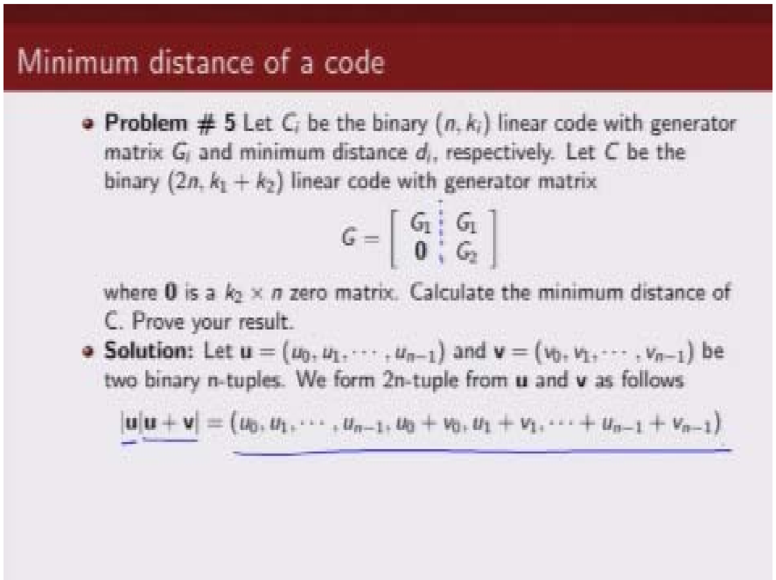
where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C. Prove your result.

- **Solution:** Let $\mathbf{u} = (u_0, u_1, \cdots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$ be two binary n-tuples. We form 2n-tuple from **u** and **v** as follows

$$|\mathbf{u}|\mathbf{u}+\mathbf{v}| = (u_0, u_1, \cdots, u_{n-1}, u_0 + v_0, u_1 + v_1, \cdots + u_{n-1} + v_{n-1})$$

So to find out the minimum distance so let us consider, let u and v are two binary n – tuples and we form a 2n tuples as follows, if you look at this code word v, how is this code word generated? So it is one n – bit code word, another n – bit code word, first n – bit code word is generated using u times t_1 and second one you get basically u times G_1 plus v times G_2 so essentially the way you are generating this code word.

(Refer Slide Time: 30:42)



Minimum distance of a code

- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i, respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \left[\begin{array}{c:c} G_1 & G_1 \\ \mathbf{0} & G_2 \end{array} \right]$$

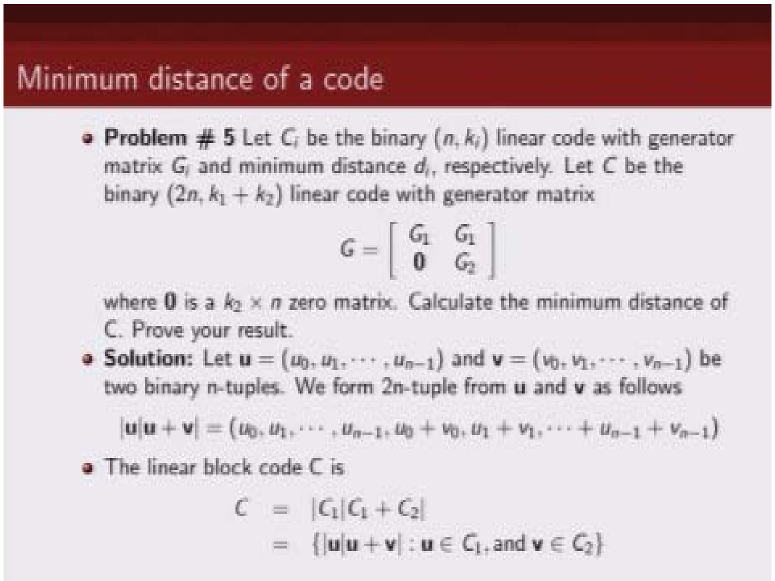
where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C. Prove your result.

- **Solution:** Let $\mathbf{u} = (u_0, u_1, \cdots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$ be two binary n-tuples. We form 2n-tuple from $\mathbf{u}$ and $\mathbf{v}$ as follows

$$|\mathbf{u}|\mathbf{u} + \mathbf{v}| = (u_0, u_1, \cdots, u_{n-1}, u_0 + v_0, u_1 + v_1, \cdots + u_{n-1} + v_{n-1})$$

The first part contains n – bit code word u and the second part is n – bit code word which is u + v, so this 2n length code word will be of the form like this where the first n - bits are u_0, u_1, u_2, u_{n-1} and the next n – bits are of the form u_0+v_0, u_1+v_1 and like that.

(Refer Slide Time: 31:13)

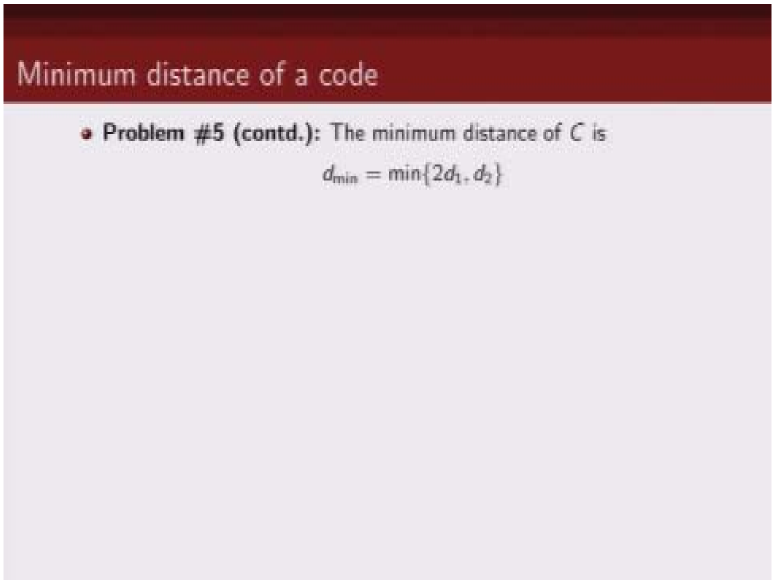


Minimum distance of a code

- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i, respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix
$$G = \begin{bmatrix} G_1 & G_1 \\ \mathbf{0} & G_2 \end{bmatrix}$$
where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C. Prove your result.
- **Solution:** Let $\mathbf{u} = (u_0, u_1, \cdots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$ be two binary n-tuples. We form 2n-tuple from $\mathbf{u}$ and $\mathbf{v}$ as follows
$$|\mathbf{u}|\mathbf{u}+\mathbf{v}| = (u_0, u_1, \cdots, u_{n-1}, u_0 + v_0, u_1 + v_1, \cdots + u_{n-1} + v_{n-1})$$
- The linear block code C is
$$\begin{aligned} C &= |C_1|C_1 + C_2| \\ &= \{|\mathbf{u}|\mathbf{u}+\mathbf{v}| : \mathbf{u} \in C_1, \text{and } \mathbf{v} \in C_2\} \end{aligned}$$

So as I said our linear block code C the new code of length 2 and can be written as this form where you have a code u of length n will belongs to C_1 and then the second part, the n - bit part is u + v where u belongs to C_1 and v belongs to C_2.

(Refer Slide Time: 31:39)



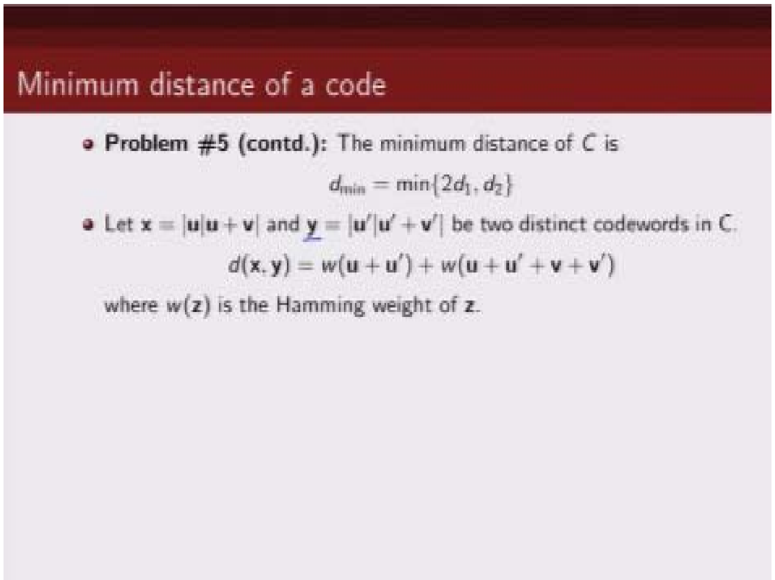
Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

Now we will show that minimum distance of the code is minimum of $2d_1$ or d_2 where d_2 is a minimum distance of the code nk_2 and d_1 is a minimum distance of a code nk_1.

(Refer Slide Time: 31:57)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

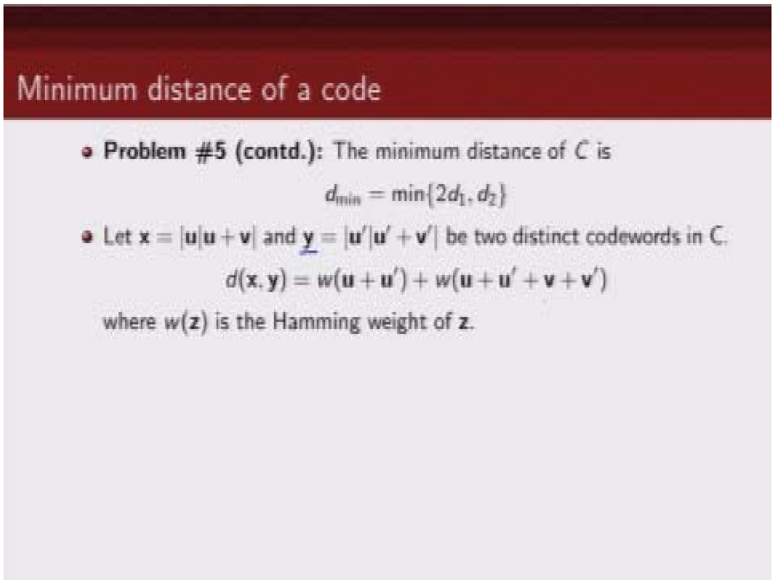
- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\underline{\mathbf{y}} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

So let us consider two distinct code word x and y, so x we denote as concatenation of u and u + v and this is u prime plus u prime plus v prime, let x and y be two distinct code words in C, now what is the hamming distance between x and y?

(Refer Slide Time: 32:29)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

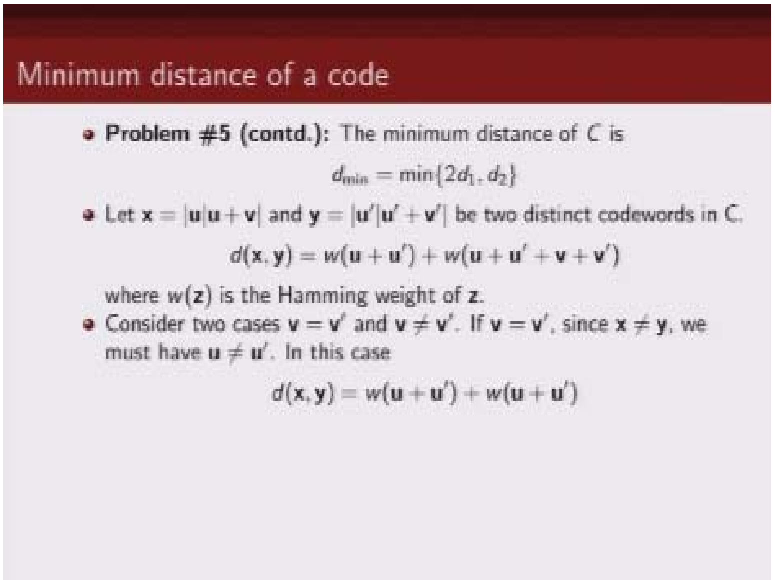
- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

We can write down the hamming distance between x and y as the hamming weight between u plus u prime plus hamming weight between u + v + u' + v' so the hamming distance between x and y can be written as hamming weight of u + u' plus hamming weight of u + u' + v + v'.

(Refer Slide Time: 32:58)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

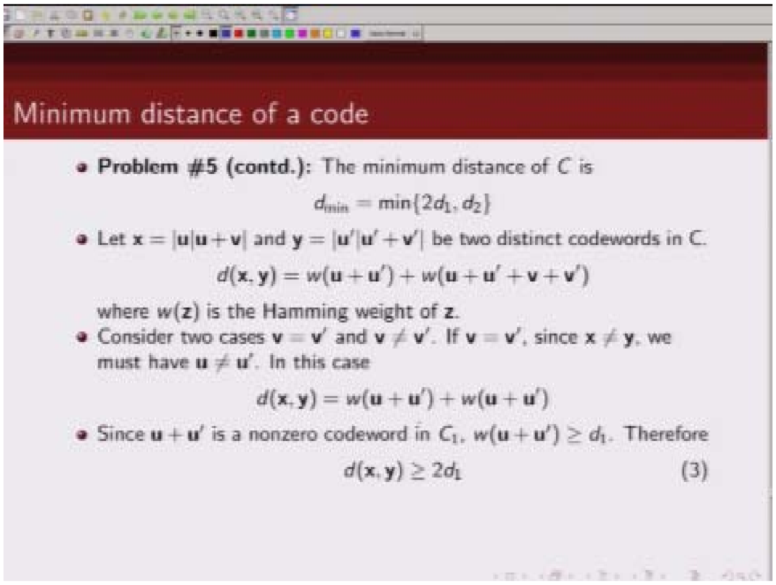
where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

Now note x and y are distinct code words so let us consider two scenarios, in first case we will consider v is same as v′ in second case we will consider v is not same as v′. So if we consider v as same as v′ since x and y are distinct code word what we will have is u is not same as u′. So in this case.

(Refer Slide Time: 33:32)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

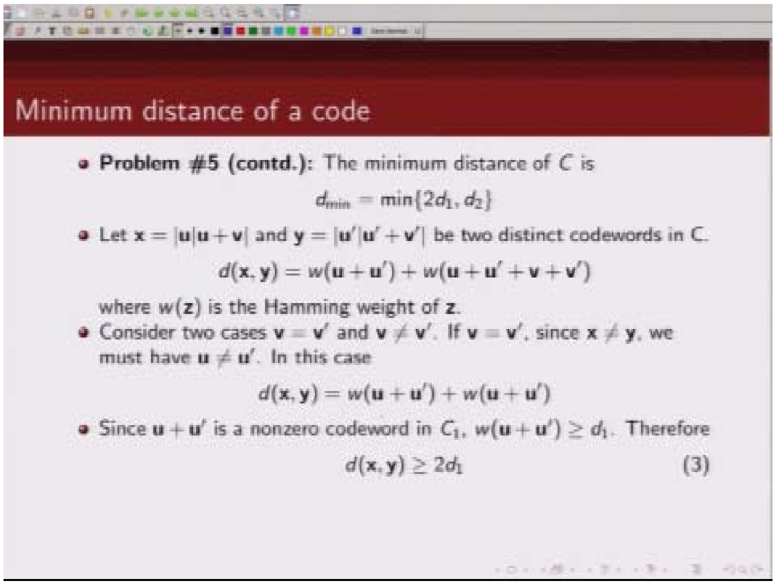
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \qquad (3)$$

The hamming distance between x and y will be given by hamming weight of (u+u′)+. Now since v and v′ are same this will be zero so this will be same as hamming weight of (u+u′). So then in this case when v is same as v′ we can write the hamming distance between x and y as hamming weight of (u+u′)+ hamming weight of (u+u′). And since what is u+u′, u and u′ are two code words belonging to C1 so sum of two code words for a linear block code is another valid code word.

(Refer Slide Time: 34:23)

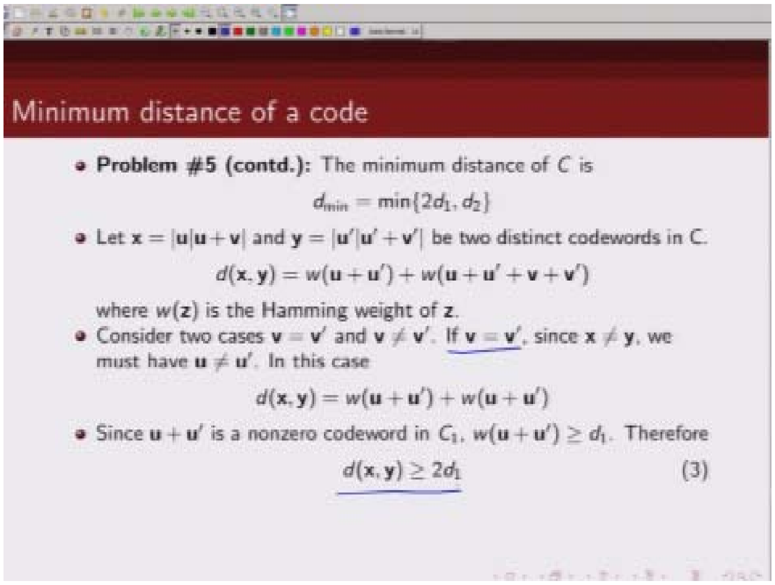


Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.
$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$
- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

So u+u′ is going to be another valid code word. So then, then what would be the minimum distance of u+u′ it would be atleast the minimum distance of the code C_1 which is d_1. So then hamming distance between x and y.

(Refer Slide Time: 34:42)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v}=\mathbf{v}'$ and $\mathbf{v}\neq\mathbf{v}'$. If $\underline{\mathbf{v}=\mathbf{v}'}$, since $\mathbf{x}\neq\mathbf{y}$, we must have $\mathbf{u}\neq\mathbf{u}'$. In this case

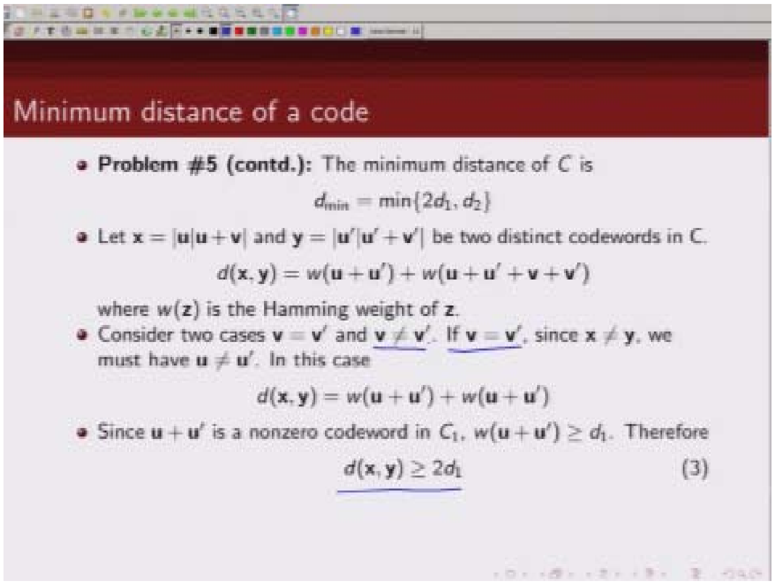
$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

$$\underline{d(\mathbf{x},\mathbf{y}) \geq 2d_1} \qquad (3)$$

Would be greater than equal to two times d1. So for the case when v=v′, we have shown that minimum distance should be atleast two times d1. Now let us consider the case.

(Refer Slide Time: 35:04)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\underline{\mathbf{v} \neq \mathbf{v}'}$. If $\underline{\mathbf{v} = \mathbf{v}'}$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

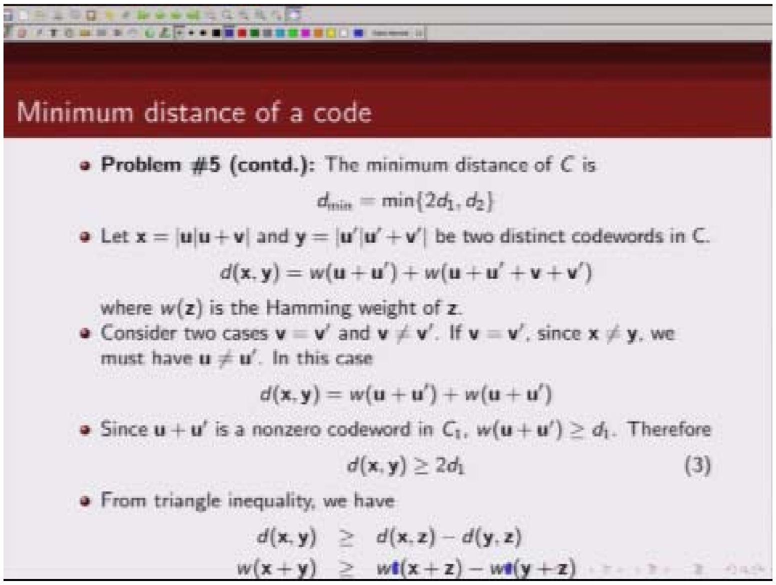
$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

$$\underline{d(\mathbf{x},\mathbf{y}) \geq 2d_1} \qquad (3)$$

When v ≠ v′.

(Refer Slide Time: 35:11)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

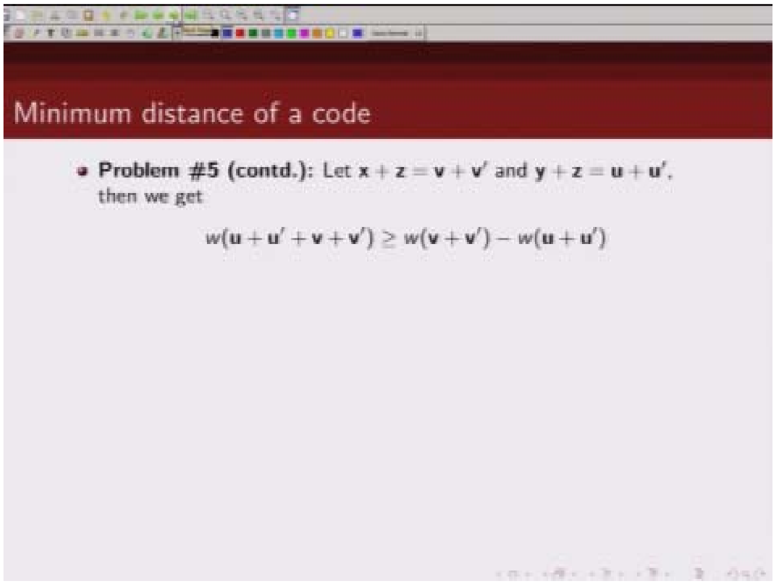
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq w(\mathbf{x}+\mathbf{z}) - w(\mathbf{y}+\mathbf{z}) \end{aligned}$$

So before that we will just state again the triangular inequalities that we are going to use, so from the triangular inequality we know that hamming distance between x and y is greater than equal to hamming distance between x and z minus hamming distance between y and z, and we know the hamming distance is nothing but hamming weight of x+y, hamming weight of x+y and hamming weight of x+z minus hamming weight of y+z. So we can write this expression in terms of hamming distance or we can write in terms of hamming weight.

(Refer Slide Time: 35:53)



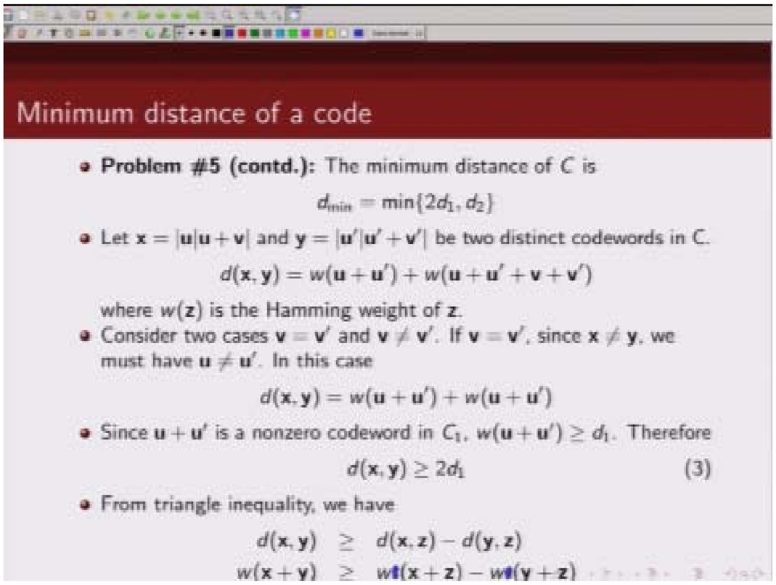
Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}') \geq w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}')$$

Now let us take x+z to be equal to v+v′ and y+z as u+u′, and we put.

(Refer Slide Time: 36:04)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

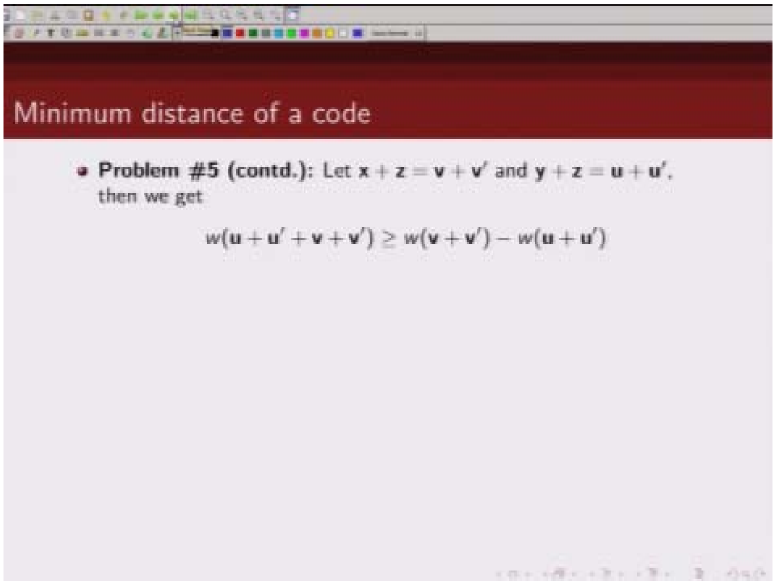
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq w(\mathbf{x}+\mathbf{z}) - w(\mathbf{y}+\mathbf{z}) \end{aligned}$$

This values of x and x+y and x+z and y+z.

(Refer Slide Time: 36:11)



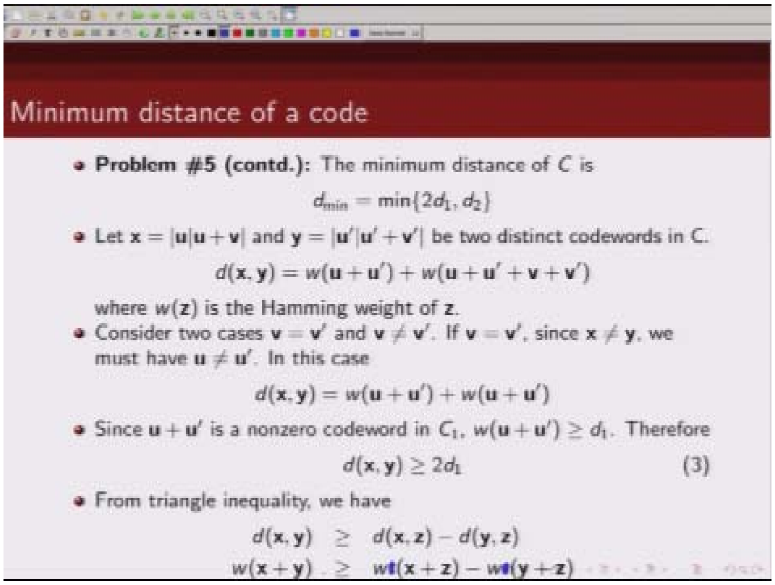
Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}') \geq w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}')$$

We put these values.

(Refer Slide Time: 36:13)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

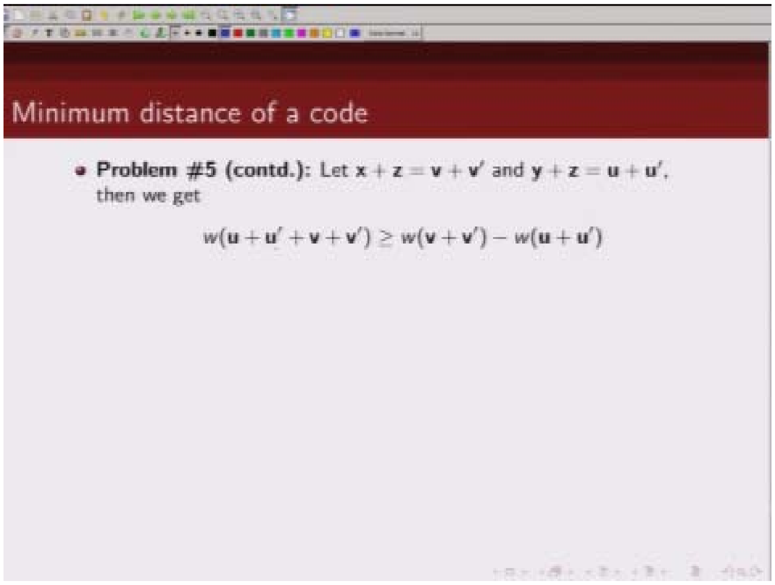
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq w(\mathbf{x}+\mathbf{z}) - w(\mathbf{y}+\mathbf{z}) \end{aligned}$$

In this expression. So what is x+y?

(Refer Slide Time: 36:18)



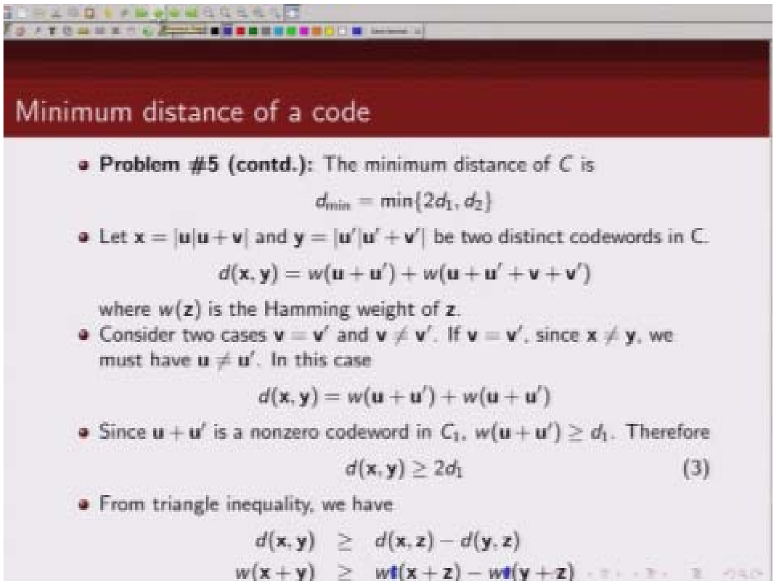
Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}') \geq w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}')$$

x+y would be v+v′+u+u′. So x+y is basically u+u′ + v+v′.

(Refer Slide Time: 36:33)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

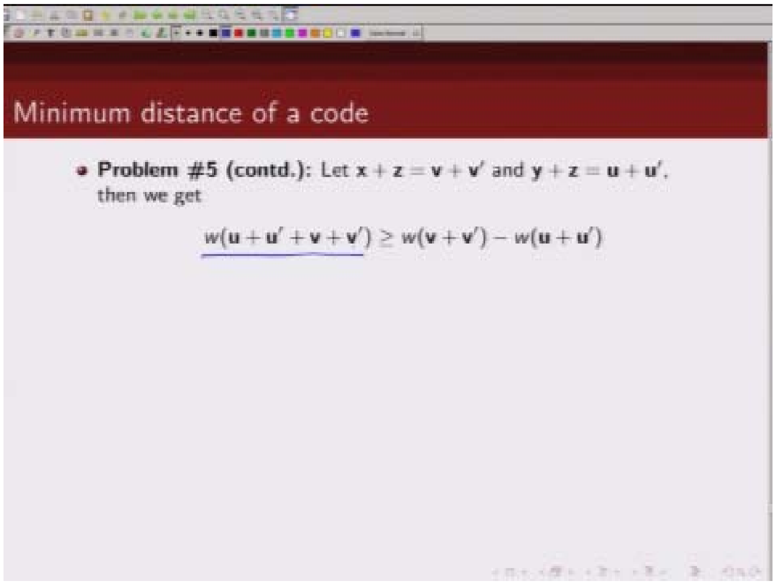
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq w(\mathbf{x}+\mathbf{z}) - w(\mathbf{y}+\mathbf{z}) \end{aligned}$$

So from here w(x+y).

(Refer Slide Time: 36:36)



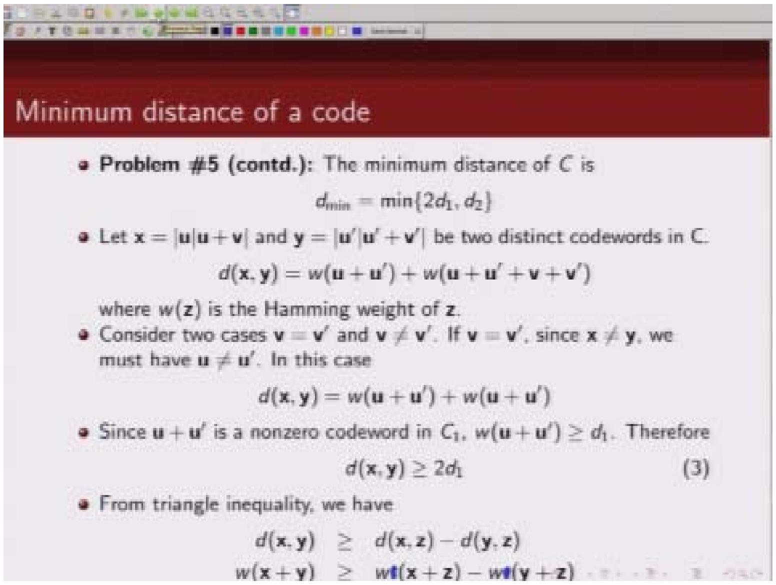
Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$\underline{w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')}\geq w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}')$$

Is given by this.

(Refer Slide Time: 36:42)

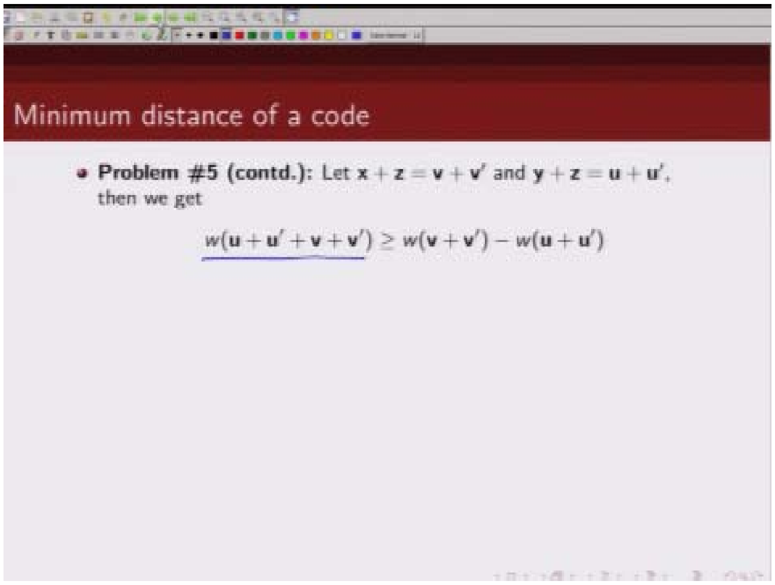


Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$
- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore
$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \qquad (3)$$
- From triangle inequality, we have
$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &\geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq w(\mathbf{x}+\mathbf{z}) - w(\mathbf{y}+\mathbf{z}) \end{aligned}$$

Next what we had was w(x+z) what is w(x+z)?

(Refer Slide Time: 36:46)



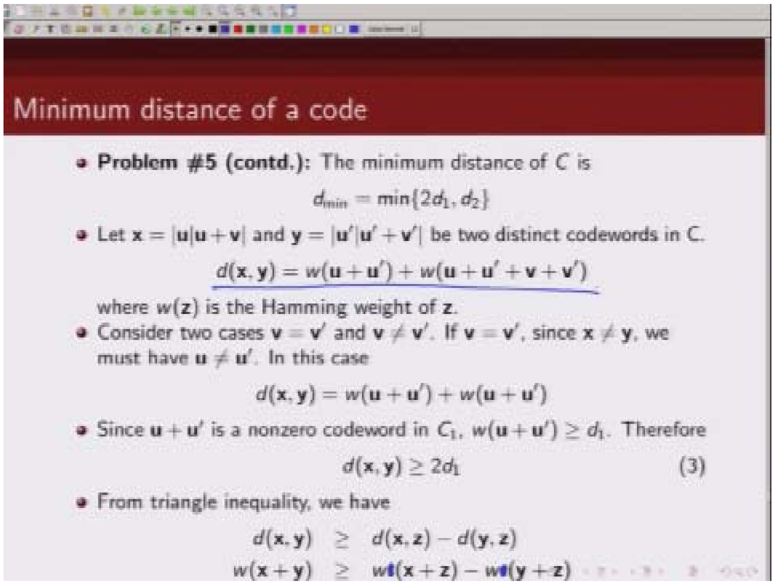
Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$\underline{w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')} \geq w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}')$$

W(x+z) is w(v+v′) and similarly w(y+z) is given by this okay. So this is upper bounded, this is lower bounded by this quantity, this is lower bounded by this quantity.

(Refer Slide Time: 37:12)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$\underline{d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')}$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

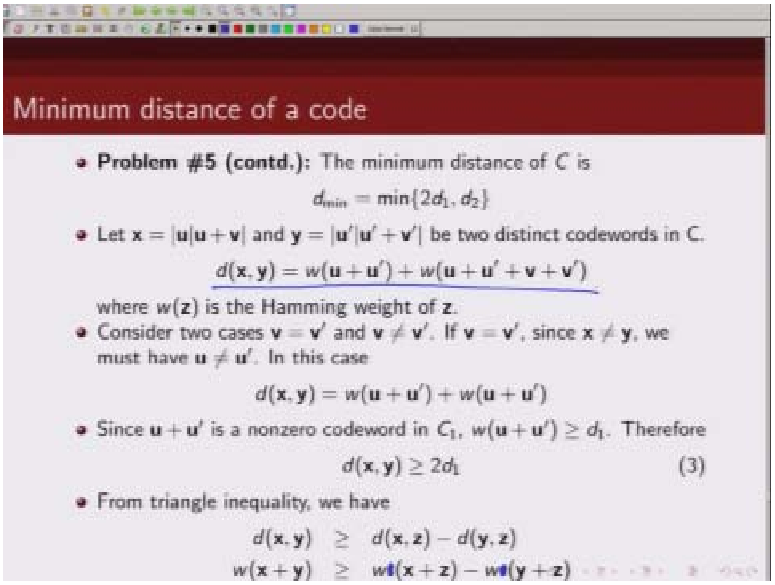
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq w(\mathbf{x}+\mathbf{z}) - w(\mathbf{y}+\mathbf{z}) \end{aligned}$$

Now go back and see what is our minimum distance between x and y, minimum distance between x and y is given by this expression, it is the hamming weight between u and u′ + plus hamming weight of u+u′.

(Refer Slide Time: 37:32)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$\underline{d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')}$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

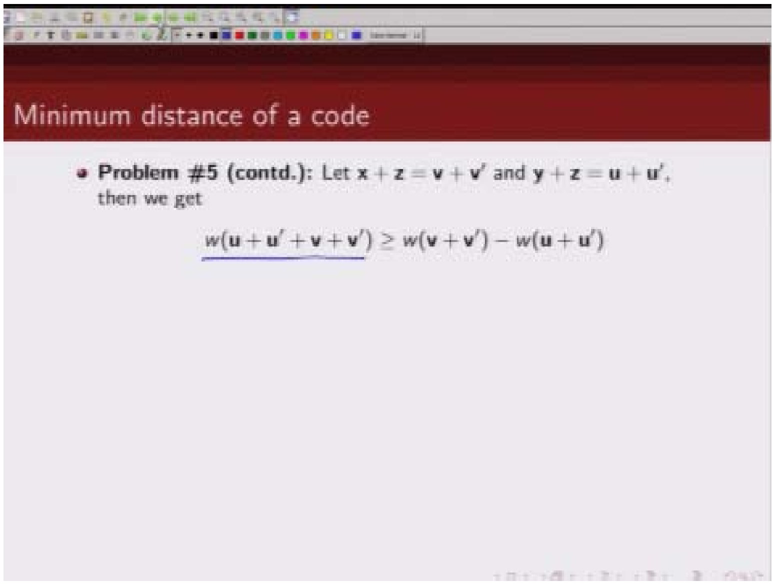
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq wt(\mathbf{x}+\mathbf{z}) - wt(\mathbf{y}+\mathbf{z}) \end{aligned}$$

Plus v+v′, and what we did just now is we lower bounded this, so then hamming distance between x and y.

(Refer Slide Time: 37:46)

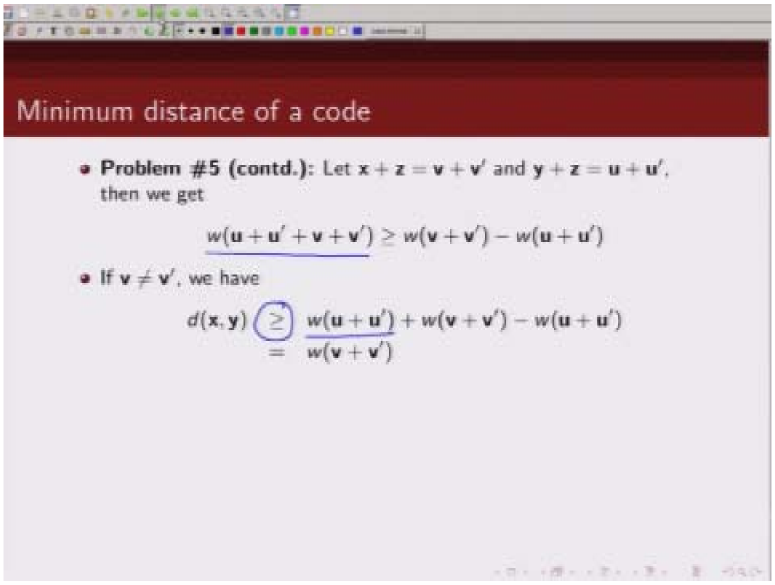


Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$w(\underline{\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}'}) \geq w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}')$$

(Refer Slide Time: 37:48)



Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

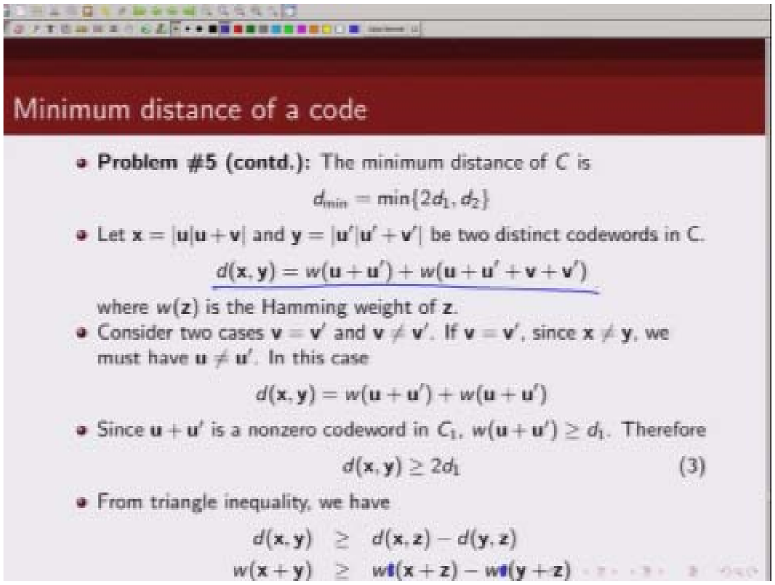
$$\underline{w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')} \geq w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}')$$

- If $\mathbf{v} \neq \mathbf{v}'$, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) & \bigcirc\!\!\!\!\geq\ \underline{w(\mathbf{u}+\mathbf{u}')} + w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}') \\ & = w(\mathbf{v}+\mathbf{v}') \end{aligned}$$

Can be, so this basically this we lower bounded by this quantity. So if you plug that in here what we get here is greater than equal to. So what we can write is the hamming distance between x and y is then greater than or equal to this term comes from here.

(Refer Slide Time: 38:11)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$\underline{d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')}$$

where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.

- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

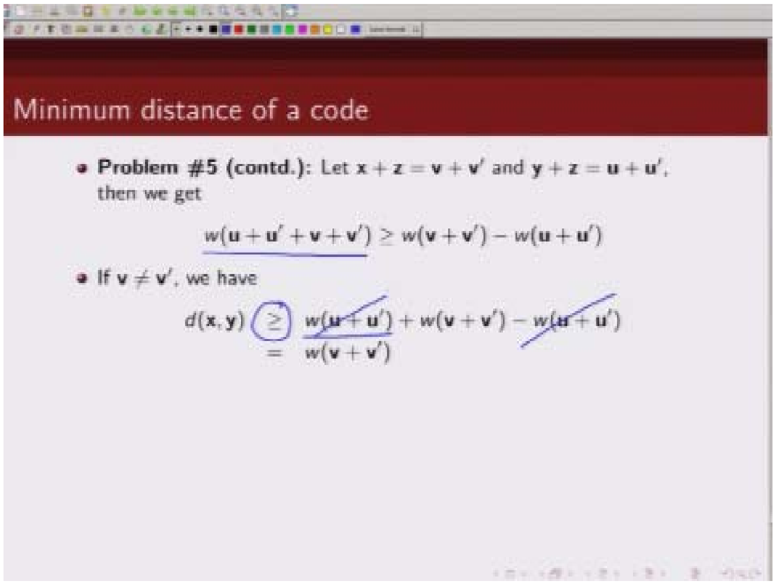
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq w\cancel{t}(\mathbf{x}+\mathbf{z}) - w\cancel{t}(\mathbf{y}+\mathbf{z}) \end{aligned}$$

This term, and this term is lower bounded by.

(Refer Slide Time: 38:15)



Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

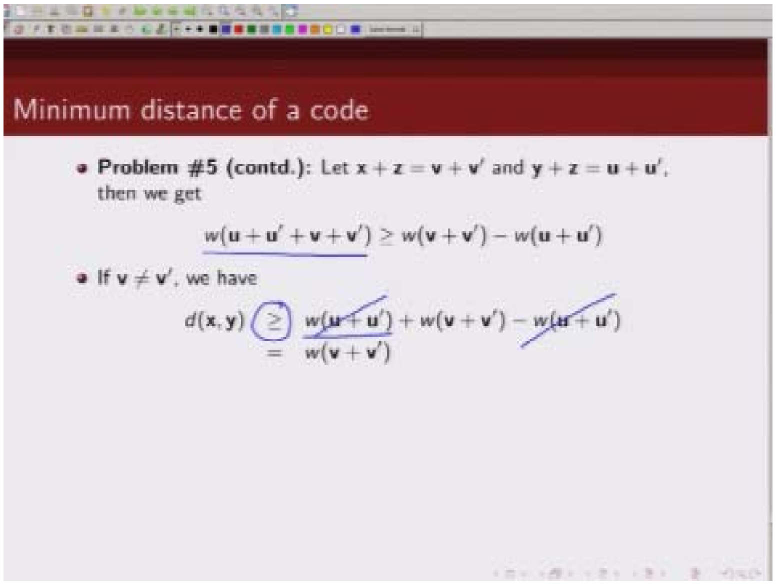
$$\underline{w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')} \geq w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}')$$

- If $\mathbf{v}\neq\mathbf{v}'$, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) & \boxed{\geq} \underline{\cancel{w(\mathbf{u}+\mathbf{u}')}} + w(\mathbf{v}+\mathbf{v}') - \cancel{w(\mathbf{u}+\mathbf{u}')} \\ & = w(\mathbf{v}+\mathbf{v}') \end{aligned}$$

This term here, so we write it here, now this can be further written as hamming weight of v+v′ because these two cancel out. So what we have shown is when v is not same as v′ the hamming distance between x and y is greater than equal to.

(Refer Slide Time: 38:42)



Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

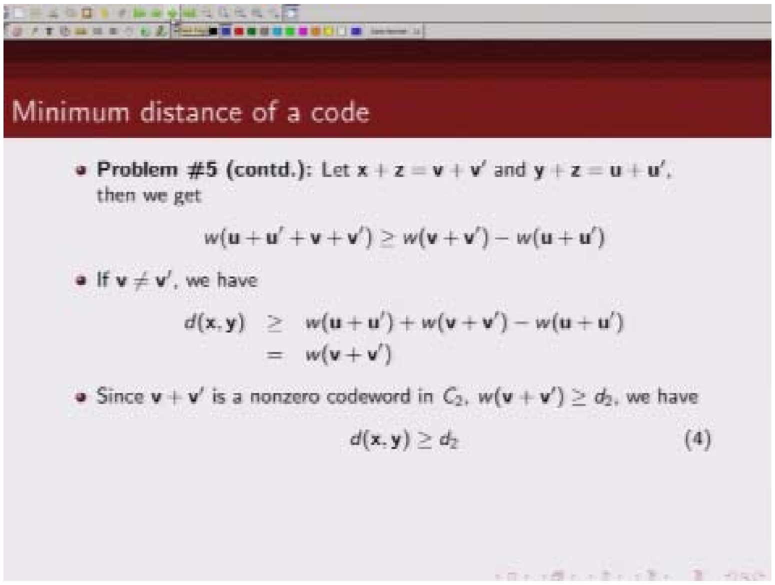
$$\underline{w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')} \geq w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}')$$

- If $\mathbf{v}\neq\mathbf{v}'$, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\boxed{\geq} \underline{w(\cancel{\mathbf{u}+\mathbf{u}'})}+w(\mathbf{v}+\mathbf{v}')-\cancel{w(\mathbf{u}+\mathbf{u}')} \\ &= w(\mathbf{v}+\mathbf{v}') \end{aligned}$$

Hamming weight of v+v′. And what is v+v′, v and v′ are valid code words in linear block code C2 with minimum distance d2. So v+v′ will be another valid code word in C2 whose minimum distance is d2.

(Refer Slide Time: 39:04)



Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}') \geq w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}')$$

- If $\mathbf{v}\neq\mathbf{v}'$, we have

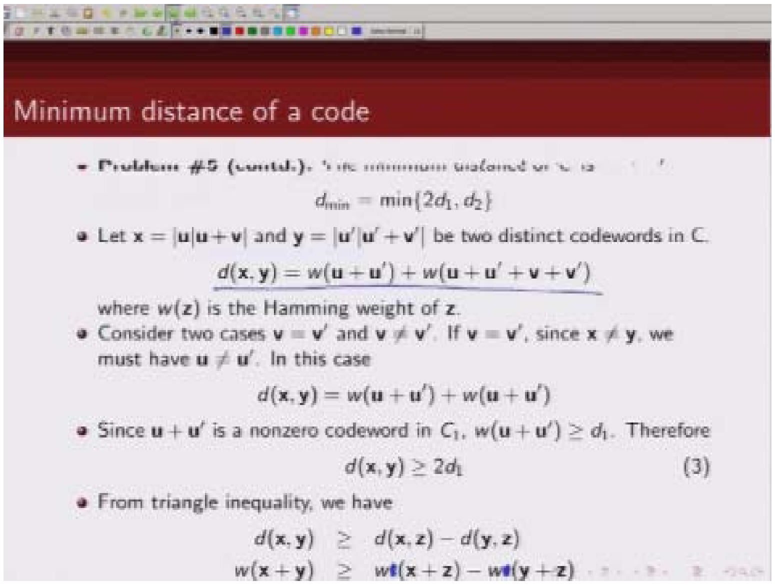
$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq w(\mathbf{u}+\mathbf{u}')+w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}') \\ &= w(\mathbf{v}+\mathbf{v}') \end{aligned}$$

- Since $\mathbf{v}+\mathbf{v}'$ is a nonzero codeword in C_2, $w(\mathbf{v}+\mathbf{v}')\geq d_2$, we have

$$d(\mathbf{x},\mathbf{y}) \geq d_2 \qquad (4)$$

So then we can write this as hamming distance between x and y is greater than equal to d$_2$. So we look comparing equation number four and.

(Refer Slide Time: 39:17)



Minimum distance of a code

- **Problem #5 (contd.).** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u}+\mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}'+\mathbf{v}'|$ be two distinct codewords in C.

$$\underline{d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}')}$$

  where $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$.
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x},\mathbf{y}) = w(\mathbf{u}+\mathbf{u}') + w(\mathbf{u}+\mathbf{u}')$$

- Since $\mathbf{u}+\mathbf{u}'$ is a nonzero codeword in C_1, $w(\mathbf{u}+\mathbf{u}') \geq d_1$. Therefore

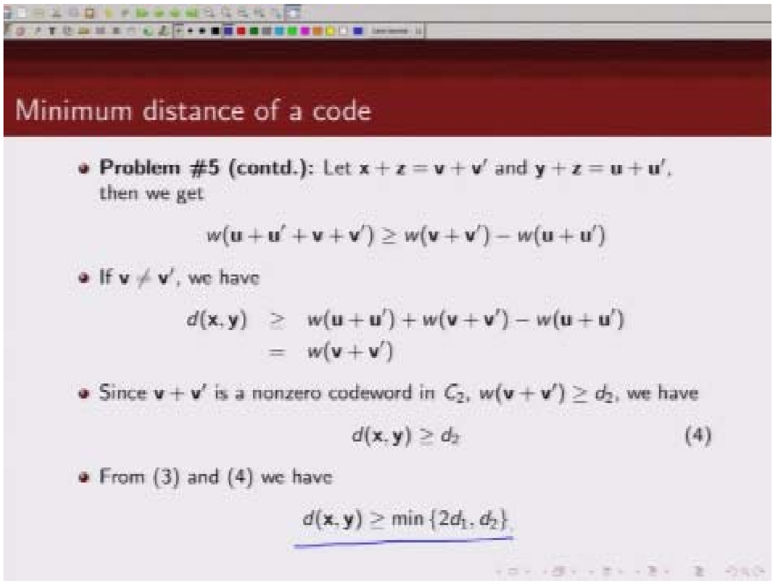
$$d(\mathbf{x},\mathbf{y}) \geq 2d_1 \qquad (3)$$

- From triangle inequality, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq d(\mathbf{x},\mathbf{z}) - d(\mathbf{y},\mathbf{z}) \\ w(\mathbf{x}+\mathbf{y}) &\geq wt(\mathbf{x}+\mathbf{z}) - wt(\mathbf{y}+\mathbf{z}) \end{aligned}$$

Equation number three, if we compare these two equations we can write that minimum distance of the code is minimum of 2d1 or d2 okay.

(Refer Slide Time: 38:33)



Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}') \geq w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}')$$

- If $\mathbf{v} \neq \mathbf{v}'$, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq w(\mathbf{u}+\mathbf{u}')+w(\mathbf{v}+\mathbf{v}')-w(\mathbf{u}+\mathbf{u}') \\ &= w(\mathbf{v}+\mathbf{v}') \end{aligned}$$

- Since $\mathbf{v}+\mathbf{v}'$ is a nonzero codeword in C_2, $w(\mathbf{v}+\mathbf{v}') \geq d_2$, we have

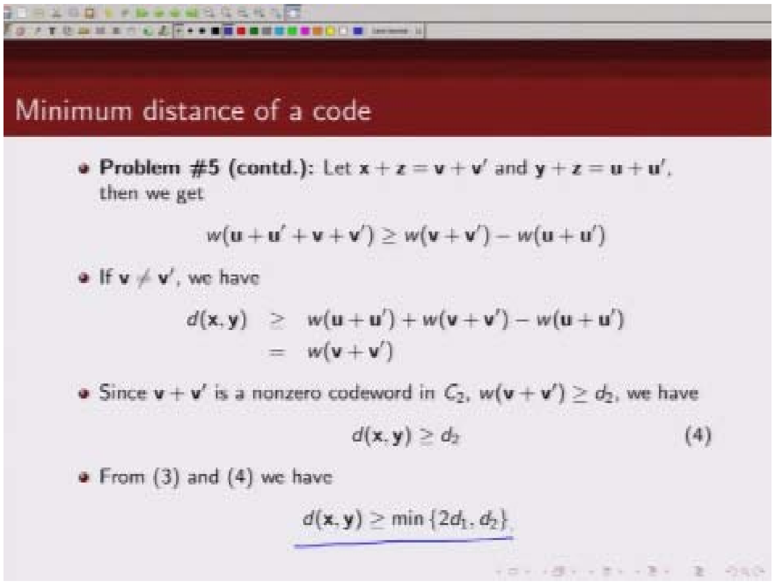
$$d(\mathbf{x},\mathbf{y}) \geq d_2 \qquad (4)$$

- From (3) and (4) we have

$$\underline{d(\mathbf{x},\mathbf{y}) \geq \min\{2d_1, d_2\}}$$

Now let us show that there exists a code with.

(Refer Slide Time: 39:42)



Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x}+\mathbf{z}=\mathbf{v}+\mathbf{v}'$ and $\mathbf{y}+\mathbf{z}=\mathbf{u}+\mathbf{u}'$, then we get

$$w(\mathbf{u}+\mathbf{u}'+\mathbf{v}+\mathbf{v}') \geq w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}')$$

- If $\mathbf{v} \neq \mathbf{v}'$, we have

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &\geq w(\mathbf{u}+\mathbf{u}') + w(\mathbf{v}+\mathbf{v}') - w(\mathbf{u}+\mathbf{u}') \\ &= w(\mathbf{v}+\mathbf{v}') \end{aligned}$$

- Since $\mathbf{v}+\mathbf{v}'$ is a nonzero codeword in C_2, $w(\mathbf{v}+\mathbf{v}') \geq d_2$, we have

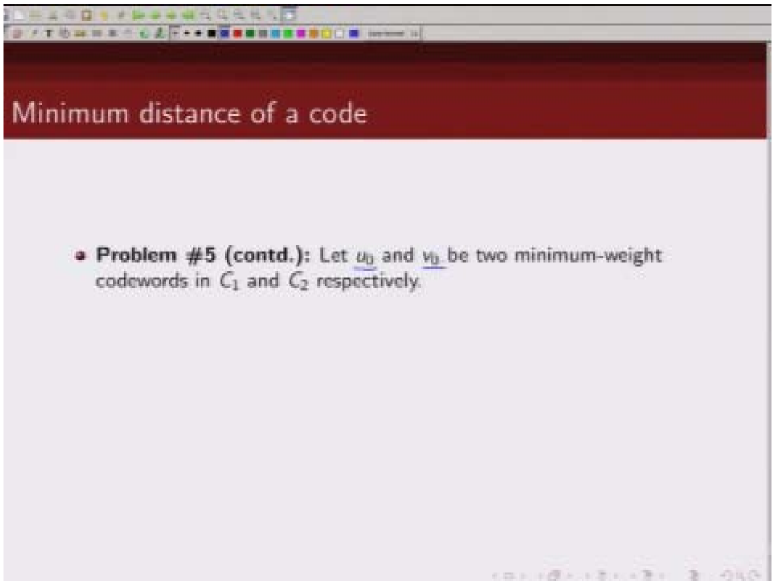
$$d(\mathbf{x},\mathbf{y}) \geq d_2 \qquad (4)$$

- From (3) and (4) we have

$$\underline{d(\mathbf{x},\mathbf{y}) \geq \min\{2d_1, d_2\}}$$

Minimum distance of code is in d equal to minimum of 2d1 or d2.

(Refer Slide Time: 39:49)

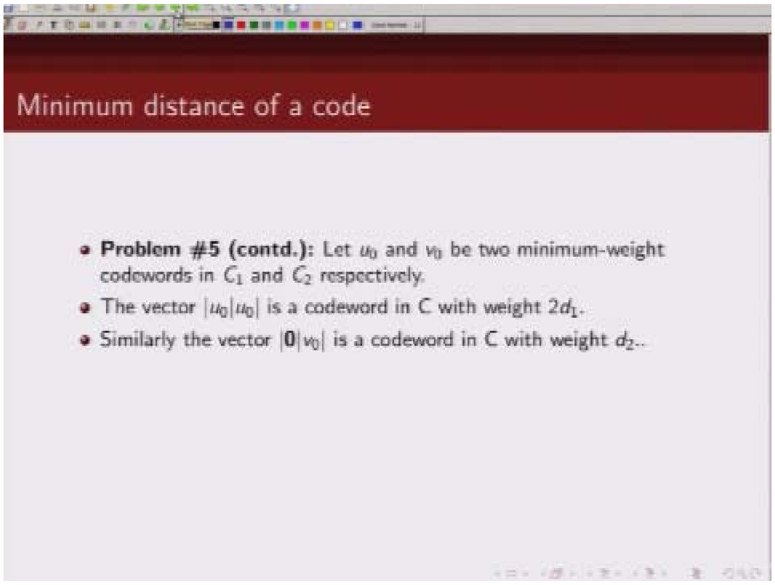


Minimum distance of a code

- **Problem #5 (contd.):** Let u_0 and v_0 be two minimum-weight codewords in C_1 and C_2 respectively.

So let us take two minimum weight code words in C_1 and C_2 let us call them u_0 and v_0.

(Refer Slide Time: 39:58)

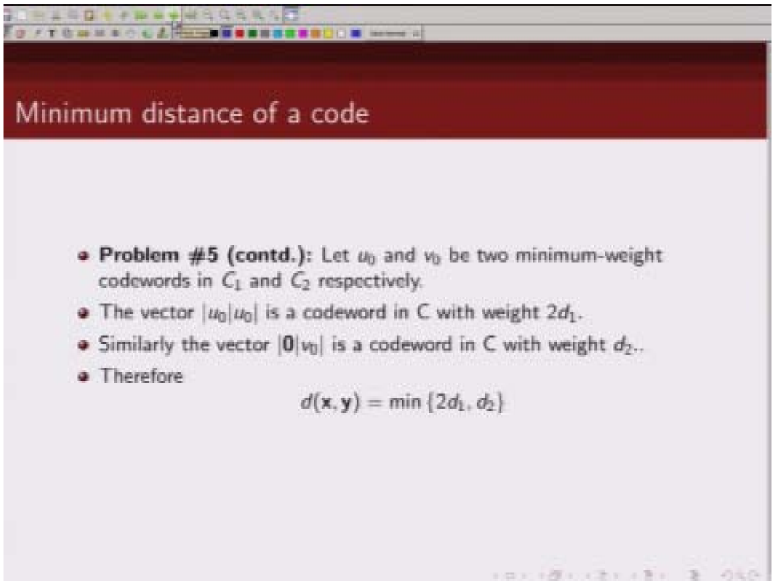


Minimum distance of a code

- **Problem #5 (contd.):** Let u_0 and v_0 be two minimum-weight codewords in C_1 and C_2 respectively.
- The vector $|u_0|u_0|$ is a codeword in C with weight $2d_1$.
- Similarly the vector $|\mathbf{0}|v_0|$ is a codeword in C with weight d_2..

Now this is a valid code word in C and what is its minimum distance, it is two times d_1 so if we take v_0 to be all zero code word what we get is u_0 and u_0 this is a valid code word in C and its minimum distance is two times d_1. Similarly, if we take u_0 to be all zero code word then what we get is this code word 0 and u_0 whose minimum distance is d_2. So hence we have shown that there basically there minimum distance of code of this code new code C is indeed minimum of $2d_1$ 0r d_2 okay.

(Refer Slide Time: 40:46)



Minimum distance of a code

- **Problem #5 (contd.):** Let u_0 and v_0 be two minimum-weight codewords in C_1 and C_2 respectively.
- The vector $|u_0|u_0|$ is a codeword in C with weight $2d_1$.
- Similarly the vector $|\mathbf{0}|v_0|$ is a codeword in C with weight d_2..
- Therefore

$$d(\mathbf{x},\mathbf{y}) = \min\{2d_1, d_2\}$$

Thank you.

Acknowledgement
Ministry of Human Resource & Development

Prof. Satyaki Roy
Co-ordinator, NPTEL IIT Kanpur

NPTEL Team
Sanjay Pal
Ashish Singh
Badal Pradhan
Tapobrata Das
Ram Chandra
Dilip Tripathi
Manoj Shrivastava
Padam Shukla
Sanjay Mishra
Shubham Rawat
Shikha Gupta
K. K. Mishra
Aradhana Singh
Sweta

Ashutosh Gairola
Dilip Katiyar
Sharwan
Hari Ram
Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari

an IIT Kanpur Production