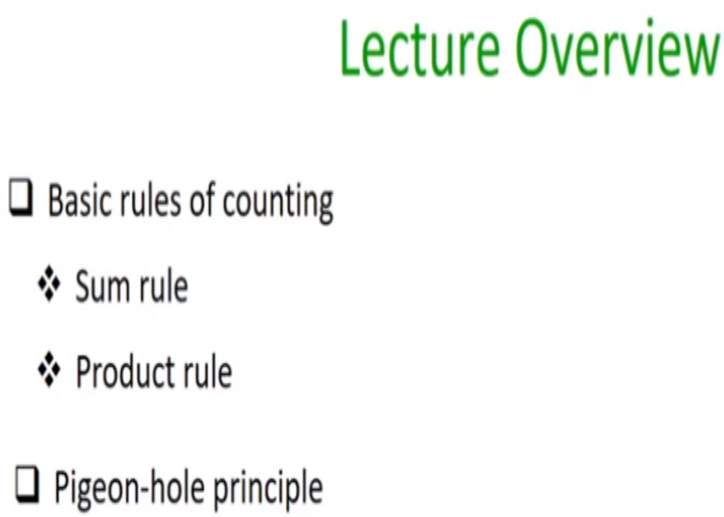
Discrete Mathematics
Prof. Ashish Choudury
Indian Institute of Technology, Bangalore

Module No # 07
Lecture No # 32
Basic Rules of Counting

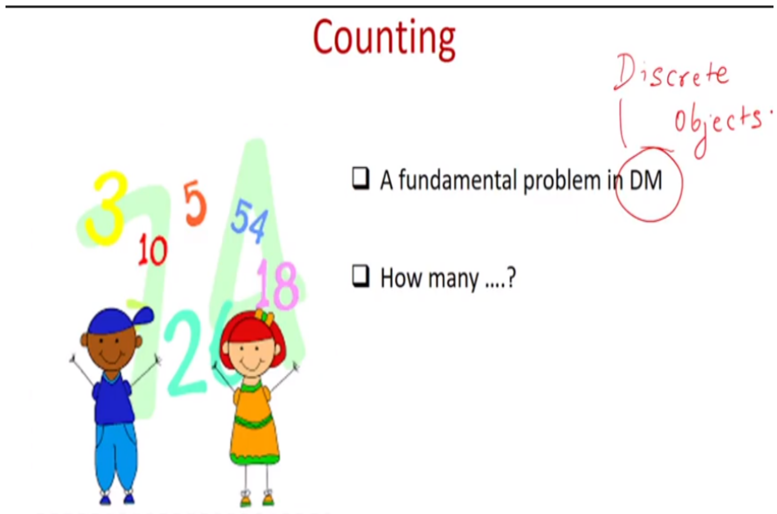
Hello everyone, welcome to this lecture. The plan for this lecture is as follows.

(Refer Slide Time: 00:27)



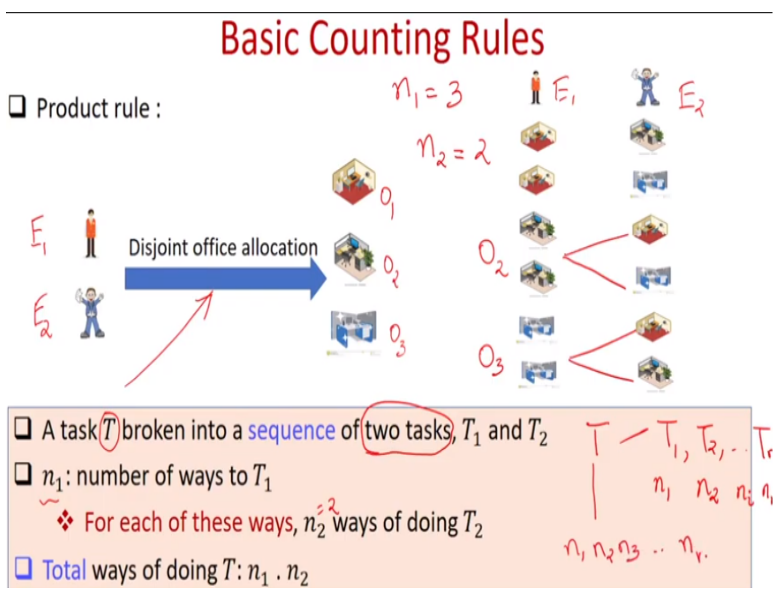
In this lecture we will introduce the basic rules of counting namely the sum rule and product rule. And we will discuss about the Pigeon-hole principle.

(Refer Slide Time: 00:39)



So we will start with the problem of counting and counting is a very fundamental problem in discrete mathematics. The reason is that in discrete mathematics we are dealing with discrete objects and since the objects that we are dealing with are discrete we can count them. So very often we will encounter questions like how many; and our main aim is to come up with methodologies to address those questions.

(Refer Slide Time: 01:15)



So we will introduce some basic counting rules in this lecture. So the first basic rule is the product rule. And I am sure all of you are aware of this product rule so let me demonstrate the product rule first with an example. So here the problem description is the following. You have 2 employees say employee number 1 and employee number 2 and they are going to join our office and there are 3 office spaces available.

So I call it office 1, office 2 and office 3; so 3 rooms are available and our goal is to identify in how many ways we can allocate disjoint offices to these 2 employees. So pictorially these are the various ways in which I can assign disjoint office to employee number 1 and employee number 2. So I can assign office number 1 to the employee 1 and given that I have assigned office number 1 to the employee 1, I cannot assign the same office to the second employee.

Because they need to be allocated disjoint offices, so I can either allocate office number 2 or office number 3 to the employee number 2. Or I can assign office number 2 to the employee 1 but in that case I cannot assign office number 2 to the employee number 2 in which case I can only assign office number 1 and office number 3 to the second employee. And similarly, I have an option of assigning the third office to the employee 1 in which case I have the options of either assigning office number 1 or office number 2 to the second employee.

So in total we have 6 ways but if you see here closely what's happening is we have a task T, a bigger task. In this example the task T was that of allocating disjoint offices to the 2 employees. And we can break that task into a sequence of 2 subtasks: subtask 1 and subtask 2. Subtask number 1 basically requires allocating office space to the first employee and subtask 2 is the problem of allocating office space to the second employee.

Suppose n_1 is the total number of ways in which we can solve the subtask 1. So in this example there are 3 ways; either I can assign office number 1 to the first employee or office number 2 to the second employee or office number 3 to the second employee. So there are 3 ways of solving the first subtask, so $n_1 = 3$ in this case and for each of these ways of solving the first subtask I have n_2 ways of solving the subtask 2.

So for instance in this example, once I have assigned office number 1 to the employee 1, I have the option of either assigning office 2 or office 3 to the second employee. So corresponding to this method of solving subtask 1; namely that of assigning office number 1 to the first employee I have

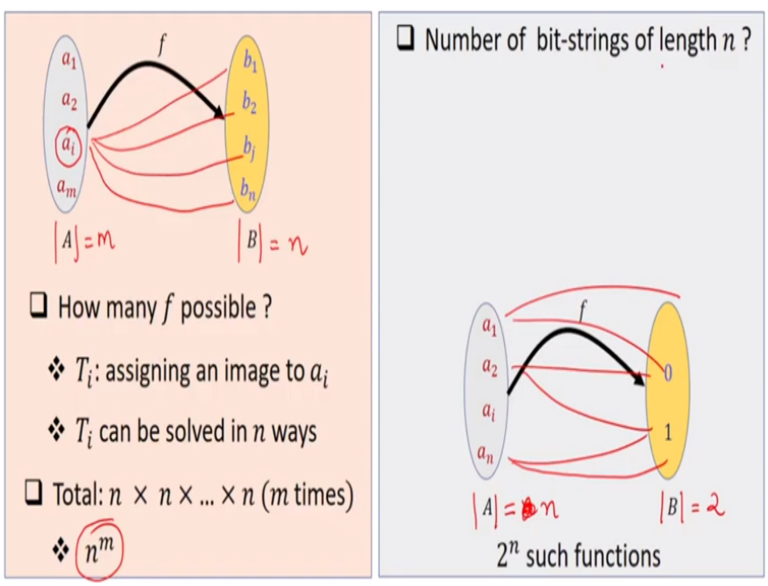
2 ways of solving subtask 2. So $n_2 = 2$ here. In the same way if I consider the method of assigning office number 2 to the second employee.

Then corresponding to this way I have 2 ways of solving subtask 2 and in the same way corresponding to the method of assigning office number 3 to the first employee. I have 2 ways of solving the subtask 2. So if this is the case then I can say that the total number of ways of solving the overall task or the bigger task is $n_1 * n_2$. And that is why in this case the answer, namely, the total number of ways of assigning the disjoint office space to the 2 employees is 6.

$n_1 = 3$ in this case, because I can either assign office number 1 to the first employee or office number 2 to the first employee or office number 3 to the first employee. So there are 3 ways and for each of these 3 ways I have 2 ways of solving the subtask 2. And that is why the total number of ways of solving the bigger task is $n_1 * n_2$. So that is the product rule. So in this case I have considered the scenario where the task T was divided into 2 subtasks.

But in general, the product rule can be applied even for cases where your task T can be divided into subtask $T_1, T_2, \dots, T_n$. So, if you have n_1 ways of solving subtask T_1 and for each of this n_1 ways you have n_2 ways of solving T_2, and for each of these ways of solving subtask $T_1, T_2, \dots, T_{i-1}$ you have n_i ways of solving subtask T_i till n ways of solving subtask T_n. Then the total number of ways of solving task T will be $n_1 * n_2 * n_3 * \dots * n_i * \dots * n_m$. That is the generalized product rule.

(Refer Slide Time: 07:46)



So now let us see some examples of product rule. So suppose we want to count the number of possible functions from a set A to a set B. My set A has m number of elements which I am denoting as a_1 to a_m and my set B as n number of elements namely b_1 to b_m. So we have already answered this question when we discussed functions. But now let us see how exactly product rule is applicable to solve this problem. So your bigger task is to find out the number of functions here.

And the bigger task is basically to assign images to each element from the set A. But now I can divide that bigger task into subtask; namely I can identify the subtask T_i which is that of assigning an image to the element a_i. And it is easy to see that the subtask T_i can be solved in n ways because if I consider the element a_i then its image can be either b_1 or its image can be b_2 or its image can be b_j or its image can be b_n.

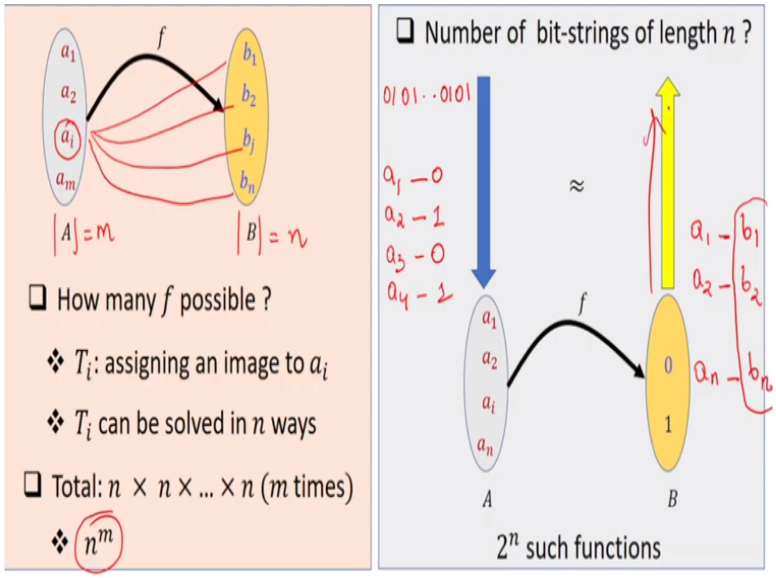
So they are n ways of solving the subtask T_i and each of this sub task are independent so that is why the total number of ways of solving the bigger task namely that of assigning image to each of the elements from the set A is $n * n * ... * n$, m number of times. And that is why the total number of functions will be n^m.

Now let's see another example. Namely we are interested to find out the total number of bit strings of length n. And there are plenty of ways to come up with an answer for this question but let us see how we can apply the product rule here. And what we will do here is instead of counting the number of bit strings of length n let's see a related problem. Namely, finding the number of binary functions; namely the number of functions from a set A consisting of n elements to a set B consisting of only 2 elements namely 0 and 1.

And from the previous exercise, here we know that the number of possible binary functions will be n^m. So the notations are actually swapped here. So we have n elements here and $|B| = 2$ so a_1 can have 2 possible images either 0 or 1, a_2 can have 2 possible images either 0 or 1, and similarly a_n can have 2 possible images either 0 or 1. So that is why we have $2 * 2 * ... * 2$, n number of times namely 2^n possible functions.

But our goal is to find out a number of bit strings of length n but what we have counted here is the number of binary functions.

(Refer Slide Time: 11:03)



So what we are now going to do is we will show here that the problem of finding the number of bit strings of length n is equivalent to finding the number of binary functions. Namely we can show that there exists an injective function from the set of bit strings of length n to the set of binary functions. And it is also easy to see that we can establish an injective function from the set of all possible binary function to the set of binary strings of length n.

And since we have established injective functions in both the directions that shows that the number of bit strings of length n is exactly the same as the number of binary functions. And the number of binary functions is 2^n. So if you are wondering what are the injective functions here, so consider you are given binary string of length n. Some arbitrary binary string of length n say $0, 1\ 0, 1$ ... like that.

Then the corresponding binary function is the following: the mapping of a_1 is 0, the mapping of a_2 is 1, the mapping of a_3 will be 0, the mapping of a_4 will be 1 and so on. That is the corresponding binary function. Whereas if you want to go from a binary function to a binary string just we do the reverse thing. So imagine you are given a binary function say where a_1 is mapped to a bit b_1, a_2 is mapped to a bit b_2 and like that a_n is mapped to a bit b_n.

Then the corresponding binary string will be b_1 to b_n. That is the injective mapping in this direction. So that shows that the number of binary strings of length n is same as the number of binary functions.

(Refer Slide Time: 13:02)

❑ Sum rule :

❑ Number of ways of forming a committee of one member --- $6 + 6 = 12$

❑ A task T can be done either in one of the n_1 ways or in one of the n_2 ways

❑ None of the n_1 ways is the same as n_2 ways

❑ Total ways of doing T: $n_1 + n_2$

Now let us consider another fundamental counting rule which is the sum rule and again let me demonstrate it first with an example. So imagine you have a set of students in a university and a set of faculty members. Of course they are disjoint because you can't have a student who is also a faculty member. And our goal is to find out the number of ways in which we can form a committee of just 1 member.

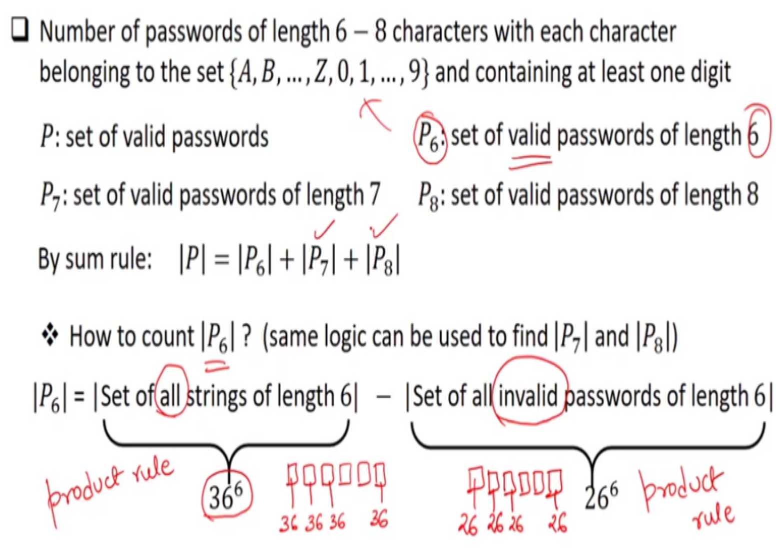
That 1 member can be either a student or a faculty. There is no restriction. We are just interested to find out how many distinct committees consisting of 1 member we can form. And it is easy to see that there are 12 ways. Why 12 ways? Because I can have a committee which consists of only a student and it could be either this student or the third student or the fourth student or the fifth student or the sixth student each of them is a distinct committee.

Or I can have a committee which has this faculty member, or this faculty member, or this faculty member, or this faculty member, or this faculty member, or this faculty member, each of which will be a distinct committee. So there are 12 different committees which we can form here. So now how we can view this as a counting rule? So the rule is the following: you have a task T which can done either in one of the n_1 ways or in one of the n_2 ways. Of course, so there is another restriction and the case here is that none of the n_1 ways is the same as the n_2 ways.

So for instance if you take this example, n_1 ways is correspond to the case when the committee consists of a student and n_2 ways correspond to the case and the committee consists of a faculty member. And both these cases are disjoint. You cannot have committee member which is simultaneously a student as well as a faculty member. So if both these 2 conditions are satisfied then I can say that the total number of ways of solving the task T is $n_1 + n_2$.

Of course in this case I have considered the scenario where the task T can be divided into 2 disjoint cases. If you have multiple disjoint cases then I can have a generalized sum rule.

(Refer Slide Time: 15:32)



So we have now seen 2 basic counting rules but it turns out that we encounter scenarios where we have to combine both these 2 rules that, means we can encounter problems which will require us to apply both the sum rule as well as the product rule. So let me demonstrate an example. So suppose we are interested to find out the number of passwords of length either 6 or 7 or 8 characters. That means the password can be either of length 6 or of length 7 or of length 8.

And the restriction is that each character can belong to the set A to Z or the numeric 0 to 9. That means the characters could be your English alphabets or digits and we also want passwords to have at least 1 digit. So these are the various requirements on the password. So it should be of length either 6 or 7 or 8. The character should be either English characters or digits and the password should have at least 1 digit.

And we are interested to find out how many such passwords we can have. So again this is a very common problem we encounter. So for instance if you consider net banking password then we have certain restrictions on the net banking password. It should be of at least this much length, it can be at most of this much length, it should have some special character etc.,

So in that case one can often ask how many such valid passwords we can form? So let us see how we can apply the sum and the product rule in this particular example. So our password, the set of all valid passwords I am denoting it as the set P and this set P actually can be divided into 3 disjoints subsets. The subset P_6 which is the set of all valid passwords of length 6; by the way by valid I mean that it has at least 1 digit and all the characters belongs to this set. That is what I mean by valid in this explanation.

So my P_6 is the set of all valid passwords of length 6, P_7 is the set of all valid passwords of length 7 and P_8 is the length set of all valid passwords of length 8 and it is easy see that these 3 sets are disjoint and by the sum rule I can say that the set of valid password, its cardinality is same as the cardinality of the set P_6 and P_7 and P_8. And there is no overlap; you can't have a password which is simultaneously of length 6 as well as length 7 as well as of length 8.

So that is why we can apply the sum rule here. Now how do we find the cardinality of the set P_6, P_7 and P_8. So let's see the logic of counting or finding the cardinality of the set P_6, the same logic is applicable to find the cardinality of the set P_7 as well as cardinality of P_8. So what exactly is the set P_6? The set P_6 is the set of all valid passwords of length 6. That means it should have exactly 6 characters, which could be either English characters or the digits, and it should have at least 1 digit.

So it can have 1 digit or it could have 2 digits or it could have 3 digits or it could have 4 digits it could have 5 digits or it could consist of all 6 digits. All these are valid passwords. So you might be attempting to apply the sum rule here but it turns out that if I apply the sum rule to find the cardinality of the set P_6 then there might be some overlaps which I have to take care off. So instead what I can do here is, I can apply the following logic. The cardinality of the set P_6 is nothing but the following.

It is the difference of the following two sets. You take the set of all strings of length 6. When I say all strings of length 6 that means they have 6 characters. But those 6 characters may or may not constitute a valid password. So for instance I may have a string of the form $AAAAAA$; 6 As belonging to the set of all strings of length 6 but this is not a valid password because it does not have a digit which is a requirement for a valid password.

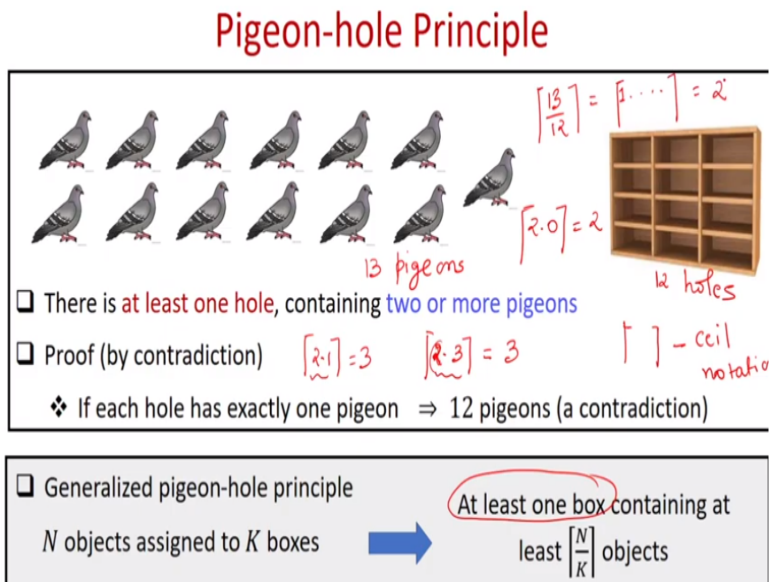
So that is why the set of all strings of length 6 have both valid passwords of length 6 as well as invalid passwords of length 6. Now from this set if I subtract the set of all invalid passwords of length 6 and by invalid passwords of length 6 I mean strings of length 6 which do not have any occurrence of a digit. Those will be the invalid password. So if I subtract those strings from this set then it is easy see that I will get the cardinality of the set P_6.

So now what is the cardinality of the set of all strings of length 6? Well it is 36^6 and this I get by applying the product rule. Why 36^6? Because I have 6 positions to fill. That means I can identify 6 sub tasks and at each position I have 36 options. I can either fill a character, English character, so 26 possibilities or I can fill any of the 10 digits. So imagine you have 6 slots here; at the first slot I have 36 options to fill, at the second slot I have 36 options, and like that at each of the slots I have 36 options.

So that is why 36^6 and what is the cardinality of invalid passwords of length 6? It is 26^6. Because here I am interested to find out in how many ways I can fill 6 slots such that none of those 6 slots is occupied with a digit. Because then only that strings of 6 characters can be considered as an invalid password. So I have 26 options now for each slot because I cannot fill any slot with a digit.

So that is why I have 26^6 options. So again here I am applying the product rule. And now if I subtract 26^6 from 36^6 that will give me the cardinality of P_6. The same logic you can apply to find out P_7 and similar logic you can apply to find out P_8. And if you sum those 3 quantities that will give you the required answer.

(Refer Slide Time: 22:57)



Now let's see another interesting counting rule which we encounter very often in discrete mathematics and this is called pigeon-hole principle. So what is the scenario here? So in this example you have 13 pigeons and you have 12 holes and suppose the pigeons are going to randomly occupy these 12 holes. We don't know in what order they will be going and occupying these holes. But irrespective of the way they are going to occupy these 12 holes we can say that there always exists at least 1 hole which will have 2 or more pigeons.

A very simple common sense. And how we can prove that? We can simply prove it by contradiction. The contradiction will be, if each hole is occupied by exactly 1 pigeon then since we have 12 holes we get 12 pigeons. But we have 13 pigeons; so that automatically implies definitely there will be 1 hole which has more than 1 pigeon. So very simple common sense here. So now how do we apply; how do we generalize this rule as a counting principle?

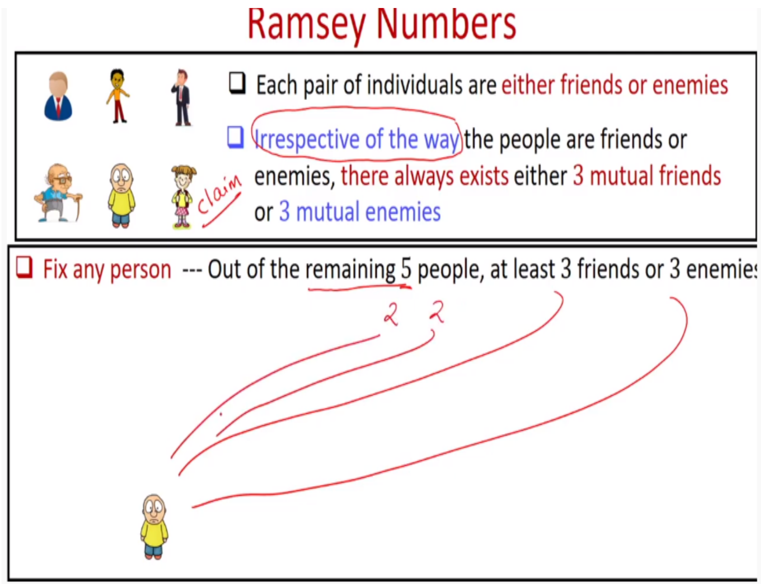
So the generalized pigeon-hole principle is the following. So imagine you have N objects, in this case you had N pigeons, and suppose those N objects are assigned to K boxes in a random fashion, then the pigeons-hole principle states that there will be at least 1 box which will have $\lceil N/K \rceil$ many objects.

So this notation is called as the ceil notation. We have $\lceil 2.3 \rceil = 3$. Basically you take the integer which is higher than the integer 2 here, that will be the ceiling of 2.2. $\lceil 2.1 \rceil = 3$; basically you take the next integer which is a complete integer and larger than the current number. Whereas the

$\lceil 2.0 \rceil = 2$ only. So in this example if I apply the generalized pigeon hole principle it basically says that there will be at least 1 box with $13/12$ pigeons and $13/12$ will be a real number.

And if I take the ceil of that I will be take the next higher integer which is 2. So the proof is by contradiction.

(Refer Slide Time: 26:14)



So let's see an application of pigeon-hole principle. So what we are now going to show is a very interesting result. So imagine you have 6 people present in a party and it is guaranteed to you that you take any pair of individuals then they will be either friends or enemies. You don't know what exactly is the situation because the party consists of 6 random people but whichever 6 random people are there in the party it is guaranteed that you take any 2 people in that party they will be either mutually friends or enemies.

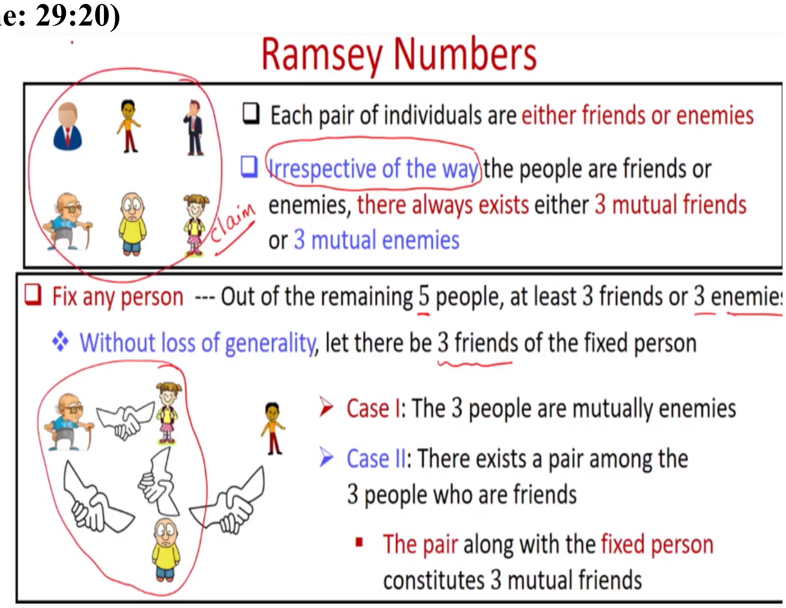
Then our claim is the following: our claim is that irrespective of the way the people are mutually friends or enemies there always exist either 3 mutual friends in the party or 3 mutual enemies. One of these 2 will definitely be the case. So how we are going to prove this? We are going to prove it by applying the pigeon-hole principle and various other proof mechanisms.

And remember, and I am making this claim; the claim is irrespective of the way of those 6 people are friends or enemies with each other. It might be the case that all of them are mutually friends then automatically the claim is true. It might be the case that none of them are friends with each other then again the claim is true. The claim is if you have 6 people definitely one of these 2 cases will always hold.

So how we are going to prove this? So we will consider an arbitrary party consisting of 6 people and out of those 6 people let's randomly choose 1 person. So we are now left with 5 people. So what can I say about those remaining 5 people. By pigeon-hole principle I can say that out of those remaining 5 people at least 3 people will be mutually friends with this person that I have chosen or there will be 3 people who are enemies, mutually enemies, with this chosen person.

I do not know what exactly is the case because that depends upon the exact way in which the persons or the people are mutually friends or enemies in the party but irrespective of the case one of these 2 will always hold. Because I have 5 people; so even if out of those 5 people say 2 are friends with this person and 2 are enemies with this person I'm left with 1 person who has to be either a friend or has to be a enemy with this person. That is a simple logic. So that is what I am saying here. So you have 2 possible cases.

(Refer Slide Time: 29:20)



So what I can say is the following: without loss of generality imagine that out of those remaining 5 people there are 3 people who are friends with this person. And since I am applying this argument without loss of generality, the same argument can be applied for the second case as well when there are 3 people who are enemies with this person. So again since there are 3 persons who are friends with this person I am taking any 3 person here who are friends with this fixed person.

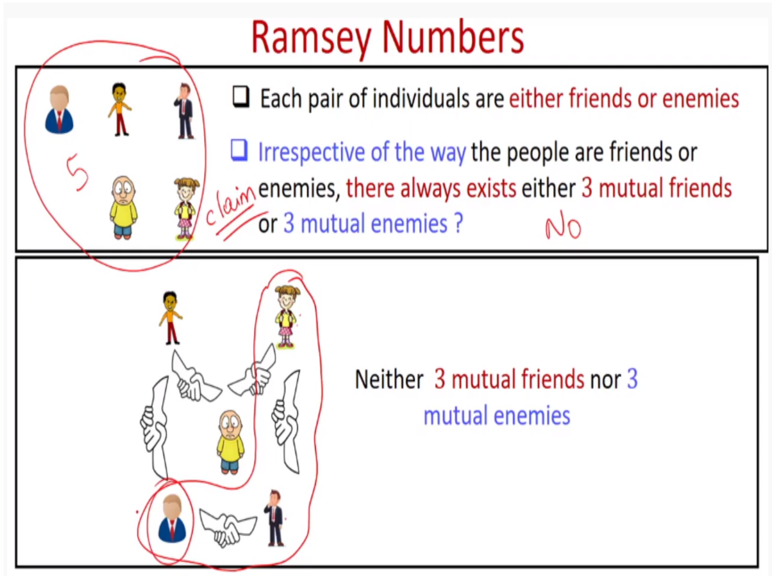
Now my claim is not yet proved here because individually these 3 people are friends with this person that does not mean that I have the existence of 3 people who are mutually friends with each other that means they all have to be friends with each other that is not guaranteed as of now. As of now I have just guaranteed that this person is a friend with this fixed person, this second person is a friend with a fixed person and the third person is a friend with a fixed person.

So this notation basically denotes friendship. Now I can say that the following 2 cases hold. The 3 people who are friends with this fixed person, they can be mutually enemies. That means, these 2 are enemies and these 2 people are enemies and these 2 people are also enemies. So let me; these 2 are enemies and this too. So if this is the case then I got 3 people who are mutually enemies with each other and my claim is true.

Whereas I can have a second case where those 3 people they are all new not mutually enemies but there exist a pair among those 3 people who are friends. Say the first 2 people are friends with each other. Then I got 3 people who are all friends with each other. This proves my claim. So now you can see I have proved my claim irrespective of the way that 6 people would have been friend or enemies.

Now the question is what is the specialty of the number 6 here. I took, I proved my claim for the case when there are 6 people in the party.

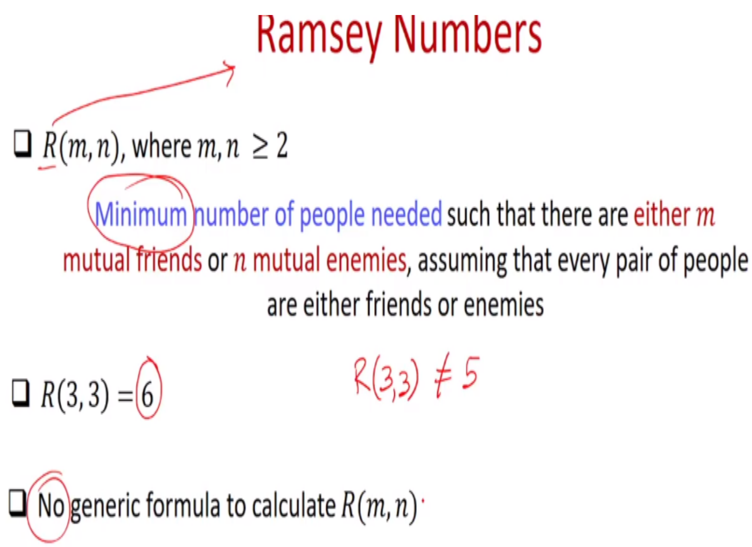
(Refer Slide Time: 32:09)



What if there would have been 5 people in the party? Then can I say that irrespective of the way those 5 people are mutually friends or enemies I will always have either 3 mutual friends or 3 mutual enemies. And answer is no. The claim is not true for the case when there are 5 people in the party. So consider the case when I have these 5 people and there is a fixed person, who is friends with this person, this person. But he is not friend with this person, who is not friend with this person and these 2 people are friend with each other and these 2 people are friend with each other so on, and these 2 people are friend with each other and so on. So in this case you can see that among these 5 people I neither have the presence of 3 mutual friends nor I have the presence of 3 mutual enemies. So for instance, if I take these 3 people then this girl is a friend with this person but that girl is not a friend with this person.

Whereas I require for my claim all the 3 people to be mutually friend or mutually enemies with each other. So when there are 5 people in the party my claim is not true.

(Refer Slide Time: 33:49)



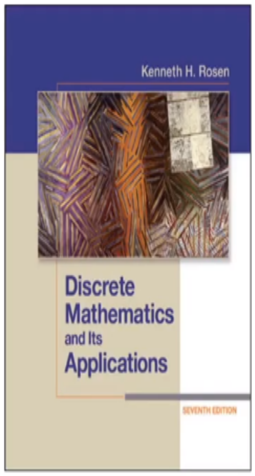
So now let us generalize this example to a beautiful theory of Ramsey numbers. So I defined this function $R(m,n)$ so this function R is attributed to Ramsey who invented these numbers and here $m,n \geq 2$. So what exactly is the value of Ramsey function $R(m,n)$? It is the *minimum* number of people required in a party such that you either have m mutual friends or n mutual enemies irrespective of the way the people are friends or enemies with each other in that party.

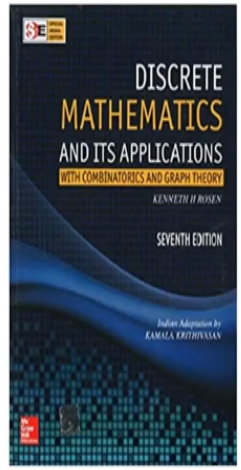
Assuming that every pair of people are either friends or enemies. So for instance what we have demonstrated is that $R(3,3) = 6$. Why 6? Because only when you have 6 people in the party then you can claim that you will either have the presence of 3 people who are all friends with each other or you will have the existence of 3 people none of them are friends with each other.

$R(3,3) \neq 5$. It is not 5 because we have given a counter example namely we can have a scenario where we have 5 people in a party such that we might have the presence of 3 mutual friends or 3 mutual enemies. So it turns out that even though this function is well defined we do not have any generic formula to find out the value of the Ramsey number or the output of this Ramsey function $R(m,n)$ for any given value of m and n. It is only for certain values of m and n that we can compute the value but there is no pattern or relationship or any observation which is there in the output of the Ramsey function due to which we do not have any generic formula.

(Refer Slide Time: 35:56)

References for Today's Lecture





So that brings to the end of today's lecture. These are references used for today's lecture. Just to summarize, in this lecture we started our discussion on counting. We introduced 2 fundamental counting rules namely the sum rule and the product rule and we also discussed the pigeon-hole principle. Thank you.