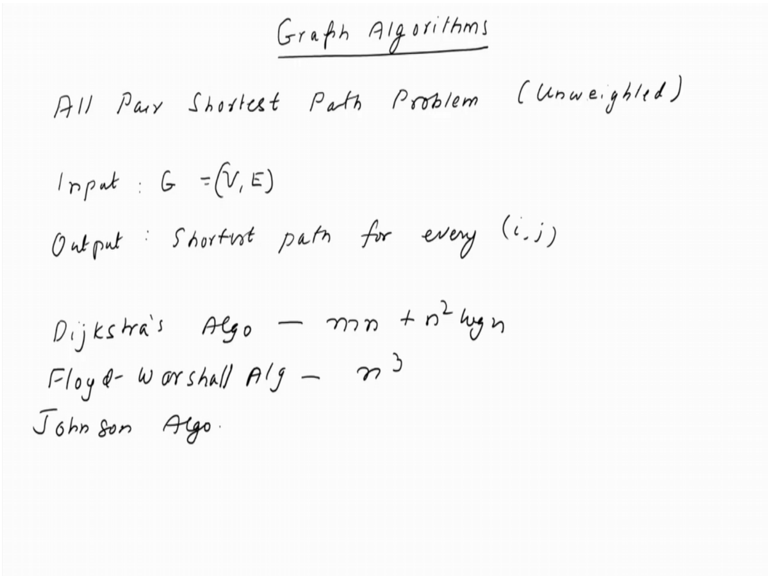
Randomized Algorithms
Prof. Benny George Kenkireth
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture - 26
All pair shortest path - I

In this lecture, we will learn about graph algorithms.

(Refer Slide Time: 00:32)

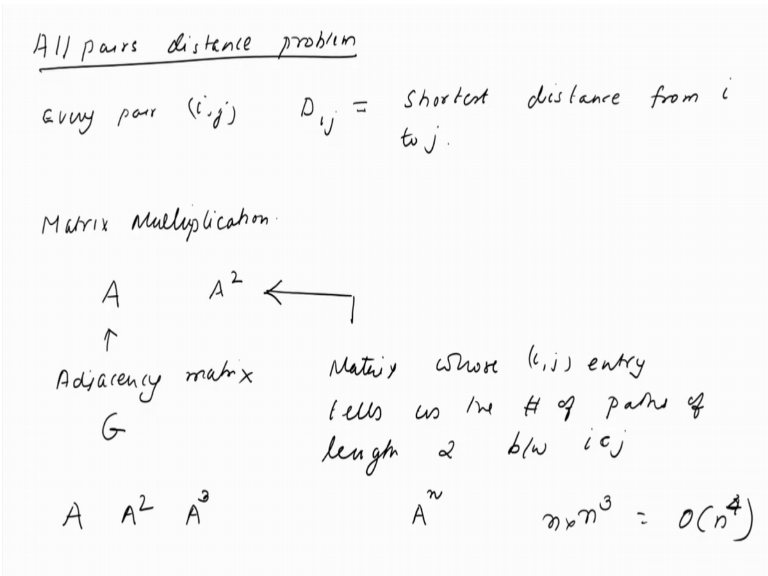


We will use how randomization can help solve the all pair shortest path problem. So, the input is a graph which is an unweighted graph and we want the output to be the shortest path between every pair of vertices. The classical algorithms, so this would be the Dijkstra's algorithm. Then there is the Floyd Worshal algorithm and there is Johnson's algorithm all these algorithm take proportional to n cube.

So, Dijkstra's can be implemented in say m times n plus n square log n time, same with Johnson's, the Floyd Worshall take n cube (Refer Time: 01:38). So, some sense you might feel like if the shortest path between pairs of vertices are long, then the output itself should take order n for every pair and their n square pair therefore n cube is the best possible. But what we will look at is a slightly modified version where in, we have given a succinct representation or an implicate representation of the shortest path.

Which would mean that, pair of vertices from this representation, we can compute the shortest path spending time proportional to the length of the path. And if we allow for the kind of output; the question is how good can we do, better than o n cube. We will see that this can be done. We will first solve a very closely related problem wherein you do not have to determine the exact path, but instead the length of the path would be fine. So, this problem we will call it has the all pairs distance problem.

(Refer Slide Time: 02:42)



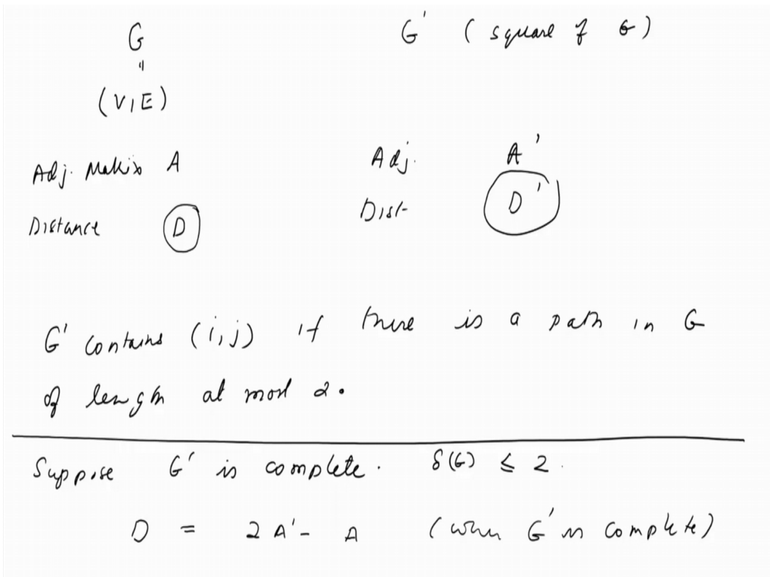
So, here the output will be for every pair i comma j, we should have matrix such that the i j’th center of that matrix is the shortest distance from i to j or shortest path; length of the shortest path ok. So, the objective is to create such a matrix. Now, this matrix the entries are all between 1 and n and therefore, or 0 and n minus 1. Therefore, the output need not be very large. How do we solve this problem? This is closely or related to the problem of matrix multiplication.

If you look at the matrix A and the matrix A square, suppose A is the adjacency matrix of some graph G , then A square is basically the adjacency matrix of a graph G prime where in, there is an edge between i and j if there is a path between i and j of distance 2. In fact, the entry will correspond to the number of such paths.

So, A square we will write it as matrix and think of it is a matrix whose i comma j entry tells us the number of paths of length 2 between i and j. So, suppose we could calculate A, A square given A if you could calculate all these different matrix products to A to the power n from these we can determine the length of the shortest path. That would basically mean that there are n matrix multiplication that you have to do and each matrix multiplication would if you use the (Refer Time: 04:53) algorithm take n cubes steps, so n into n cube it has much more than cubic.

The question here would be, since there is some relation to matrix multiplication. Can we use matrix multiplication without doing all these calculations and still come up with the all pair shortest distances? So, we will do that. Let us introduce a few terms and we will look at a square of a graph.

(Refer Slide Time: 05:26)



So, let us say G is a graph and G prime is another graph which we will refer to as square of G ok. So, we will define what it means to call a graph is a square of another graph. Some sense the adjacency matrices squared, but you also allowing for paths of length 1 ok.

So, if G is the graph denoted by V comma E and let A be is adjacency matrix ok, let D be it is distance matrix ok. When we talk about the square of graph, it means a graph which is obtained by adding all those edges such that, there were either connected in G or connected by an edge or they were connected by a path of length 2. In other words, G prime contains i comma j as an edge, if there is a path in G of length at most 2 ok, i comma i edges we do not bother about it ok.

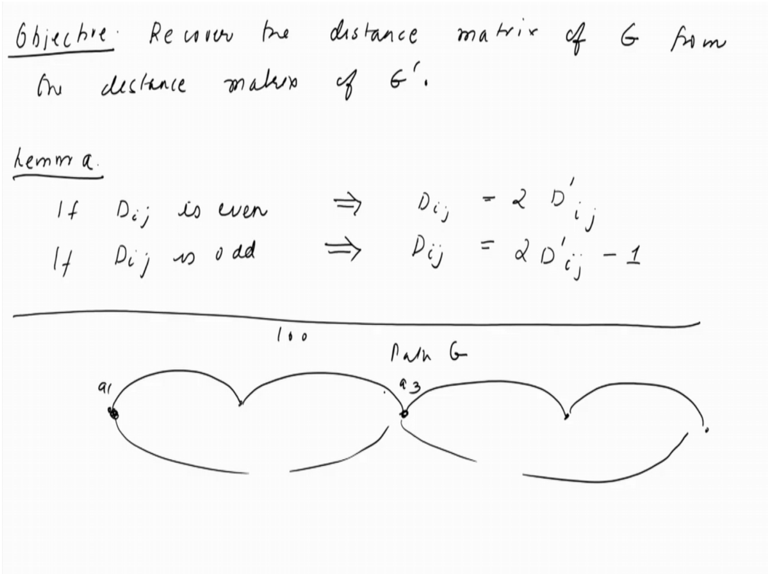
So, that is our graph G prime. Now the adjacency matrix of G prime let us call that as a prime and the distance matrix let us call it as D prime. A couple of observations based on this definition, if you look at a path in G of length lets a 100, surely g prime will contain a path between these two vertices itself of length 50. If there was a path of length 101, then G prime would contain a path of length 51 ok.

We will make these notions formal ah. So, let us just look at this statement when suppose, G prime is complete ok; it is a complete graph, this will happen when is G prime a complete graph. This happens only if the diameter of G is going to be less than or equal to 2 means every vertex was relate was connected to another vertex by a path of length at most 2. If that is a case, then g prime is going to be complete and the 3 if an only if condition.

Now, if G prime was complete, can we somehow use D prime to compute D ok, are they related? So, we have this particular relation D is 2 times A prime minus A when G is G prime is complete ok. So, if you know the adjacency matrix of G prime twice that minus A would give you G. So, that is a matrix computation.

So, from the distance matrix we can figure out that it is a complete graph. The square of G is a complete graph. Then from that information, we can recover back the distance matrix for G by taking twice the adjacency matrix of A and subtracting A from it. This works only if the diameter is 2. For arbitrary diameter what can we do. So, can we somehow look at the square of the graph and from the distance matrix of the square of the graph recover back the distance matrix of the graph ok.

(Refer Slide Time: 09:24)

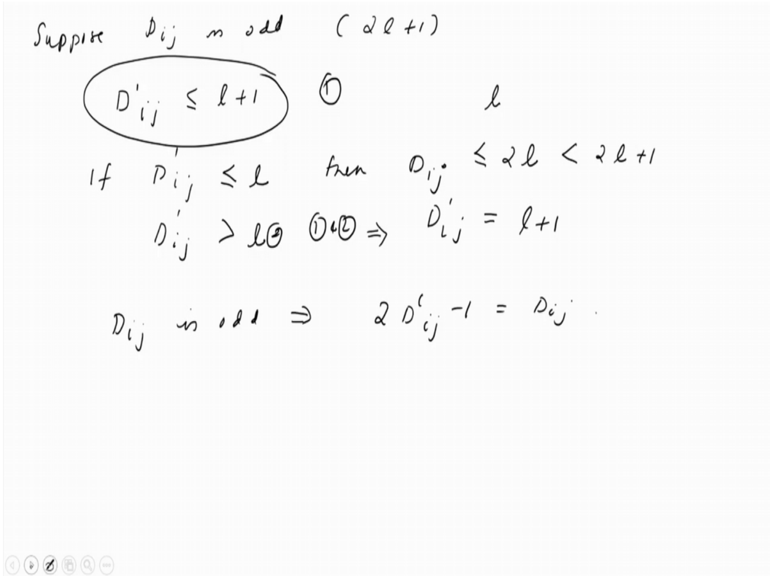


So, that is our objective. Recover the distance matrix of G from the distance matrix of G prime. Why are we doing this? Suppose we want to compute in our objective was to compute the distance matrix of G, instead we will compute the distance matrix for G prime and that we will do recursively. At some stage the square will be the complete graph and for the complete graph we know how to compute the distance matrix of its square root and we can basically used that procedure to have a recursive algorithm to solve the all pair distance problem.

So, this is useful lemma that we will use. If D ij is even, where D ij denotes a distance between i and j in graph G. This would imply that, D ij is equal to 2 times D prime ij ok. So, D prime i j is the distance between i and j in this square squared graph. And if D ij is odd, this would imply that D ij is equal to 2 times D prime ij minus 1. So, let us see what is an easy statement. Why is this 2?

So, let us look at any path. If this path was of length 100 and this is the path in G, automatically and so this is a path G, you can skip one vertex when your traversing this in prime, because there is an edge form say a 1 to a 3 and a 3 to a 5 and so on ok. So, any path of length 100, we will have a corresponding path of length 50 in G prime ok. And if the path is of length 101, then there will be surely a path of length 51 in the other graph ok. So, let us see a formal proof of this.

(Refer Slide Time: 12:11)



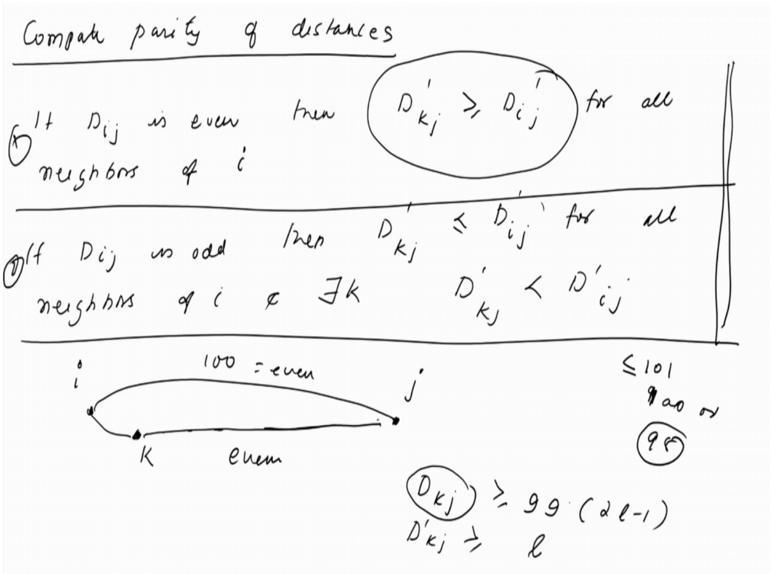
We will just do one case where D ij is odd. So, suppose D ij is odd, so it is of the form 2 l plus 1 ok. Then clearly D prime ij, so D prime ij is the shortest path between i and j in this square graph. That is going to be less than l plus 1, less than or equal to l plus 1 because take 2 at a time, the last edge has to be used as it is, the other one's contribute well last one contributes 1. So, l plus 1 is the length of the shortest path, when is no more than l plus 1. If you look at this path between i and j in the graph G prime, clearly there is a path of length at most l plus 1, can it be anything less than l plus 1 ok.

So, suppose it is l or any other number smaller, so let us just take l. If it is l then, you can take such a path in G prime and convert it into a path of length at most 2 l. In other words, it is a first part D ij is less than l plus 1, the second part would be if D ij prime is less than or equal to l, then D ij will be less than or equal to 2 times l which is strictly less than 2 l plus 1.

But we assume that the shortest path in G was of between i and j is of length 12 plus 1, therefore D i j cannot be less than or equal to l. In other words, D ij should be greater than l and that is precisely one number. So, this would imply, so 1 2 would imply that D prime ij is equal to l plus 1. So, when D ij is odd, the following relation holds this would implied 2 times D prime i comma j minus 1 is equal to D ij.

So, what does this tell us? We have a graph whose distance is we already know for example, G prime. Means distance matrix is denoted by D prime. From D prime, we can recover D i j. If we know the parity, the oddness and evenness of the distances, but now the question becomes I mean would be to determine the parity of these distance; if you know the parity, then we can recurs and compute all the individual distances ok.

(Refer Slide Time: 15:00)



So, how do we compute parity? Ok, in the following easy lemma helps us do this. If D ij is even, then D kj prime is greater than D i j prime for all neighbors of i. And if D ij is odd, then D kj prime is going to be less than D ij prime for all neighbors of i. And there will exist a k such that D prime kj is less than D prime ij ok.

So, we given a condition by which ascertain whether D ij is even or odd. So, let us see why such statement is true. So, let us first consider the case where D ij is even, the shortest path from i to j is of even length. If you look at the shortest path from k to j, so this is i and this is j. If you look at the shortest path from k to j, it is either passing through I, if it is passing through i, surely let us going to be greater than or equal to, or it is a path of odd length ok. So, here this is even and k to j it passes through i mean i.

This cannot be the k to j path cannot be of even length, because if it were of even length then i to j shortest path would be of odd length which contradicts are assumption that i to j shortest path was even ok. Suppose, k to j shortest paths of length say some even length.

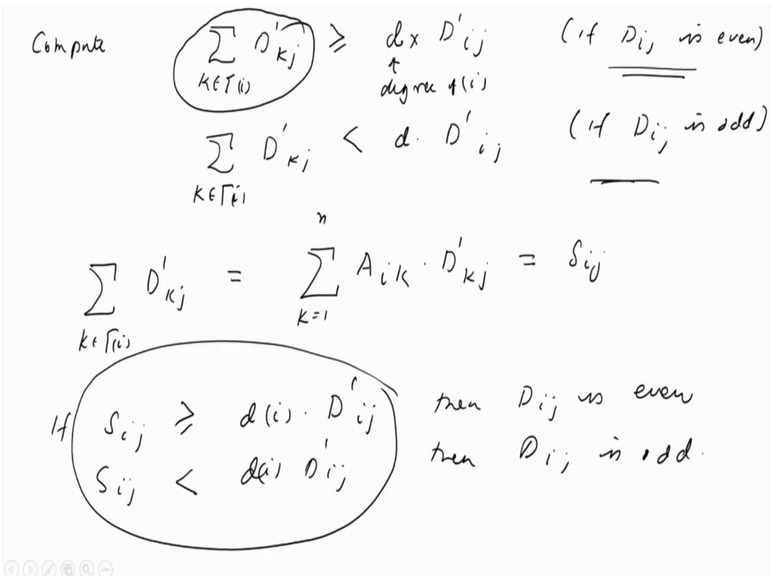
Now, this shortest path from k to j its length is certainly less than or equal to 101, because there is surely a path of length 101 and if we additionally assume that this k to j path; shortest path was of even length then it can only be 90 then it can be 100 or when if it is 98, there is a problem because if it is 98 then there is a path from i to j whose length is only 100.

So, if the i to j path is even that would mean that D k j for any neighbor this is going to be greater than or equal to 99 or this will be greater than 2 l minus 1 ok. And therefore, by the previous lemma, we can conclude that D prime k j is going to be greater than D prime ij is just going to be half. So, that is going to be 50 or l and D kj is going to be greater than or equal to l ok. So, that is the proof that whenever D ij is a even, there would surely we i mean for all neighbors of i D prime kj is greater than or equal to D i j prime.

And the similar argument works for the case where D ij is odd. When D ij is odd, you can show that D prime kj is going to be less than D prime ij for all neighbors and there where surely be one neighbor for which this is strictly less ok. So, these characterizes the nature of the parity of D ij.

So, we could look at neighbors in the squared graph and look at their distance and based on whether they are greater than or less then we can decide whether D ij is even or odd. So, you can write this as a simpler condition and cleaner criteria.

(Refer Slide Time: 19:20)



So, compute summation D prime kj, where k varies over neighbors of i. This should be greater than d times D prime ij, if D ij is even. And this summation k belonging to neighbors of I, D prime kj should be strictly less then d into D prime ij if d i j is odd.

So, if you wanted to know whether D ij is odd or even, we basically need to compute these quantities. That is, summation k belonging to neighbors of i D prime kj. And if it is greater than the degree of i times D prime i j, then we can conclude that it is even otherwise, we can conclude that it is odd. And this calculation, summation k belonging to neighbors of i D prime kj can be written as a matrix multiplication. Is this product over A all k going from 1 to n a ik ok. So, A ik is 1 only if k is a neighbor of i times D prime kj. So, this if you call it as S ij.

Then become compute these quantities, if S ij is greater than d i into D prime i j, then D ij is even and if it is less, then D ij is odd. So, by doing 1 matrix multiplication, we can determine the parity of D ij. This we can convert this into an algorithm we write down the algorithm, so the deterministic algorithm to determine the distances.

(Refer Slide Time: 21:56)

Input : $G = (V, E)$
Output: Distance matrix of G.

1. $Z = A^2$
2. Compute A' from Z & A.
3. If $A'_{ij} = 1$ for all i, j $(\neq j)$, then
   $D = 2A' - A$ (return D)
4. Compute the distance matrix of A'. (D')
5. $S = A \cdot D'$
6. $D_{ij} = 2D'_{ij}$ $(S_{ij} \geq D'_{ij} \cdot Z_{ii})$ # of neighbors
   $= 2D'_{ij} - 1$ $(S_{ij} < D'_{ij} \cdot Z_{ii})$

So, we will start of with an the input is a graph and the output will be the distance matrix of G. So, first we will have to compute Z is equal to A square, so this is the adjacency matrix of the graph obtained by taking where there is an edge with an i and j, if and only if that is a path of length 2 between i and j. Then compute A prime, the adjacency matrix of this square of G from Z and A ok, so that would involve little bit of calculation.

In A prime, there is an edge between i and j. If there is either a path of length 2 or a path of length 1 ok and i is not equal to j; so, A prime can be computed. And if when A prime ij is equal to 1 for all i comma j where i not equal to j, then d will be equal to 2 times A prime minus A. Then we will return this particular D. This is if you have computed the adjacency matrix of the square and if it is the complete matrix, then we readily compute the distance matrix from the adjacency matrices and return that. So, that is not the case, we need to recursively compute the distance matrix of A prime.

So, compute the distance matrix of A prime ok so, that is let us call it as D prime and it is adjacency matrix is A prime. We will compute S is equal to A times D prime, computing A times D prime, so that we can do this check which helps is determine whether D ij is even or odd and from S we are going to compute D. So, D ij is equal to 2 times D prime ij. This is when D ij was even and how do we check whether D ij is even, you have to look at the S ijth entry and if that is greater than D prime ij times Z ii. In Z ii was a square of A if there is a path between i and i of length 2, so the condition that we needed to check here is S ij is greater than or equal to D i times D prime ij.

So, D prime i j is available here and that multiplied by the degree. So, the degree would be just the i ith entry in the square. So, this is the number of neighbors ok and this is equal to 2 times D prime ij minus 1 if S ij is less than D prime ij into Z ii ok.

So, this is how we compute the all pair distance matrix ok. Now, the question becomes, if we have the all pair distance matrix can we use that to compute the shortest path that is what will be the next topic.