Course: Introduction to Graph Algorithms

Professor: C Pandu Rangan

Department: Computer Science and Engineering

Institute: IISc

Week: 05

Lecture 20 All Pair Shortest Path 2

Namaskara, we have seen an order n power 4 algorithm based on Bellman's equations for bounded paths and we will see how it can be improved, okay? There is another interesting way to look at the computation of Dl from Dl minus 1. This view allows us to build a more efficient algorithm. This computation resembles the multiplication of two matrices. It will be little surprising for you that a computation that is done using minimum and addition in getting the matrix Dl from Dl minus 1, can indeed be viewed as a multiplication of matrices, okay? We are going to show the similarity and that will give you an insightful way more interesting way to look at the computation that are done in obtaining Dl from Dl minus 1, okay. Suppose we interpret the symbol plus for minimum operation instead of sum, that is a plus b is minimum of a and b. For example, 5 plus 3 is minimum of 5 and 3 which is 3.

$$a + b = \text{Min}\{a, b\}$$

$$5 + 3 = \text{Min}\ \{5,3\} = 3$$

Under this interpretation, the usual interpretation will be sum, which is 5 plus 3 is 8, but now I am interpreting plus as a symbol, just an abstract symbol, that stands for min operation. It takes two input but it outputs the minimum rather than the sum. So if this is interpreted then the minimum of n numbers a1, a2, an, can be written simply as sum of i going from 1 to n of ai. Notice that here the Sigma is not integer sum, but finding the minimum, okay? So the resemblance.

$$\sum_{i=1}^{n} a_i = \text{Min}\ \{a_1, a_2, \cdots, a_n\}$$

You can see how the, for the sum operation 0 works as an identity element, 5 plus 0 is 5. What is the identity element when plus is interpreted as minimum? In other words, minimum of a and whatever it is should be a for all a, it is infinity, that is because minimum of if you look at minimum of a and infinity, is a.

$$\text{Min}\ \{a, \infty\} = a$$

$$a + \infty = a$$

So in our notation a plus infinity, is a, because plus stands for minimum, a plus infinity is a so you can see that infinity is acting like an identity element. The association is associative, the operation is associative, the operation is commutative, therefore all the properties that the plus has for some min also has that property. So whatever you have done in that interpretation you can carry it out under this interpretation also, okay.

We will make one more operation and interpret differently. Now dot is interpreted as plus, for example 5 dot 7 is 5 plus 7 which is 12 under the new interpretation.

$$5 \cdot 7 = 5 + 7 = 12$$

So the dot which is usually for multiplication I am now going to use it for addition. Plus I am interpreting for minimum, dot I am interpreting for addition. Let us see how the equation dlij, this is a Bellman equation, under the new interpretation it gets transformed, okay? You can see step by step transformation.

First of all, the plus is replaced by dot because in my new notation plus is nothing but a dot and the minimum is nothing but the plus which is sum. Therefore, dl ij can be written as, sigma k equal to 1 to n dl minus 1 ik.wkj, okay?

$$d_{ij}^{l} = \sum_{k=1}^{n} d_{ik}^{l-1} \cdot w_{kj}$$

Under the new interpretation, the same expression but with new symbol just new symbol is written like this. Now the resemblance to matrix multiplication is clearly visible, for example look at the formula for matrix multiplication. If C equal to AB where A is an n by n matrix, B is an n by n matrix, C is the n by n matrix where cij, cij is given by Sigma k goes from 1 to n of aik and bkj.

$$C = AB$$

$$A = \left[a_{ij}\right]_{n\times n}$$

$$B = \left[b_{ij}\right]_{n\times n}$$

$$C = \left[c_{ij}\right]_{n\times n}$$

$$\text{where } c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

So you can see that k goes from 1 to n here, k goes from 1 to n here, and the element of the matrix A here element of one matrix here and the element of another matrix is here and this is the element of another matrix. So here Dl is Dl minus 1W because this is a entry of matrix W.

$$D^{l} = D^{l-1}W$$

This is like this is an entry of matrix A, this is the entry of matrix B, here I have a formula, this is an entry of matrix D l minus 1, this is the entry of matrix W, therefore I am going to write Dl as Dl minus 1 W, look at a new way of looking at the computation that is done. It resembles, matrix multiplication. So everything that you have intuitively built and understand for matrix multiplication can literally be applied in this context, absolutely no issues on this.

Therefore, you write D1 as D0w, D2 as D1w which is D0w times w. Matrix multiplication is associative, moving the w to the other side you get d0 w square and so on.

$$D_1 = D_0 w$$

$$D_2 = D_1 w = (D_0 w)w = D_0 w^2$$

therefore, we have Dn minus 1 is D0 W power n minus 1.

$$D_{n-1} = D_0 W^{n-1}$$

This is the power of the matrix W power n minus 1, so you can see that instead of going through, starting from D0 apply one matrix multiplication and get D1, apply another matrix multiplication and get D2. If you go like this, each matrix multiplication is n cube operation and this is going to take n power 4. However, if we can directly find W power n minus 1 there is a matrix, there is a power of the matrix you have to find, right.

Instead of multiplying W times W times W is there a way I can quickly get W power. For example W power 32 by repeated squaring, we can quickly go in 5 steps from W to W square, square that matrix you get W power 4, square that you get W power 8, square it W 16 square it.

$$W \longrightarrow W^2 \longrightarrow W^4 \longrightarrow W^8 \longrightarrow W^{16} \longrightarrow W^{32}$$

Therefore, squaring is multiplying the matrix with the same matrix therefore, when you keep doing that you can see that W power 32 is obtained in 1, 2, 3, 4, 5 steps. In general, W power 2 power k can be computed in k squaring steps. Here is the idea behind a more efficient algorithm.

Since the longest path can have only n minus 1 edges Dt is equal to Dn minus 1 for all t. Any t larger than n minus1 will have the same value. Therefore, Dt can be computed, Dn minus1 can be computed by computing a Dt which is the nearest to next power of 2. For example if you want to compute D power 13, we know that D power 13 is same, sorry yeah, D power 13, we are interested in computing W power 13, but I am not going to compute W power 13, I will compute W power 16 in 4 steps, but W power 16 is same as W power 13, right. Therefore, I am going to compute D power 16 like this and D power 13 is same as D power 16.

$$D^{(16)} = D_0 W^{16}$$

$$D^{(13)} = D^{16}$$

Since we have this equation we are able to get it done using log n multiplications, okay? This is our second algorithm, I keep doing this log n times, seal log n times, okay?

$$l = 1 \text{ to } \lceil \log n \rceil$$

$$W = W^2$$

I take to the next power of 2, whatever is n like if I want to compute W power 13 I will compute instead of this W power 16, if I want to compute W power 97 I will compute W power 128. The next highest power of 2. So W power t when I do this I will have W for some t greater than n minus 1.

$$W = W^t \text{ for some } t > (n-1)$$

Therefore, D is D0W therefore D is D power t for the same t whatever is the t I found but Dt is Dn minus 1 and this is what we want therefore I am returning D very simple proof

$$D = D^{(0)} W$$

$$D = D^{(t)}, \text{ for the same } t > (n-1)$$

$$\text{Hence } D = D^{(t)} = D^{(n-1)}$$

And the complexity is also direct each squaring takes n cube time. and you are doing only log n plus 1 maximum, seal log n okay.

$$(\log n + 1) n^3$$

Therefore the complexity is order n cube log n which is better than n power 4

$$O(n^3 \log n)$$

Just by looking at the same computation in a different perspective just by seeing the computation that we do is resembling the matrix multiplication okay. We converted the problem of computing Dn minus1 as D0W power n minus 1 this is matrix power.

$$D^{(n-1)} = D^{(0)}W^{(n-1)}$$

This matrix power can be computed in log n steps by the doubling or squaring process keep squaring it you quickly reach the power of W that is the key idea behind this improvement okay. This concludes our discussions on algorithm 2 our first algorithm was taking n power 4 steps the same equation when we looked at as something similar to matrix multiplication has led to an improvement in the efficiency we have significantly cut down the complexity from n power 4 to n cube log n. and this completes our discussions on algorithm 2. Thank you.