CS & IT ENGINEERING

Algorithm

Algorithms

Lecture No. 17

By- Ravindrababu Ravula Sir

Recap of Previous Lecture

Topic

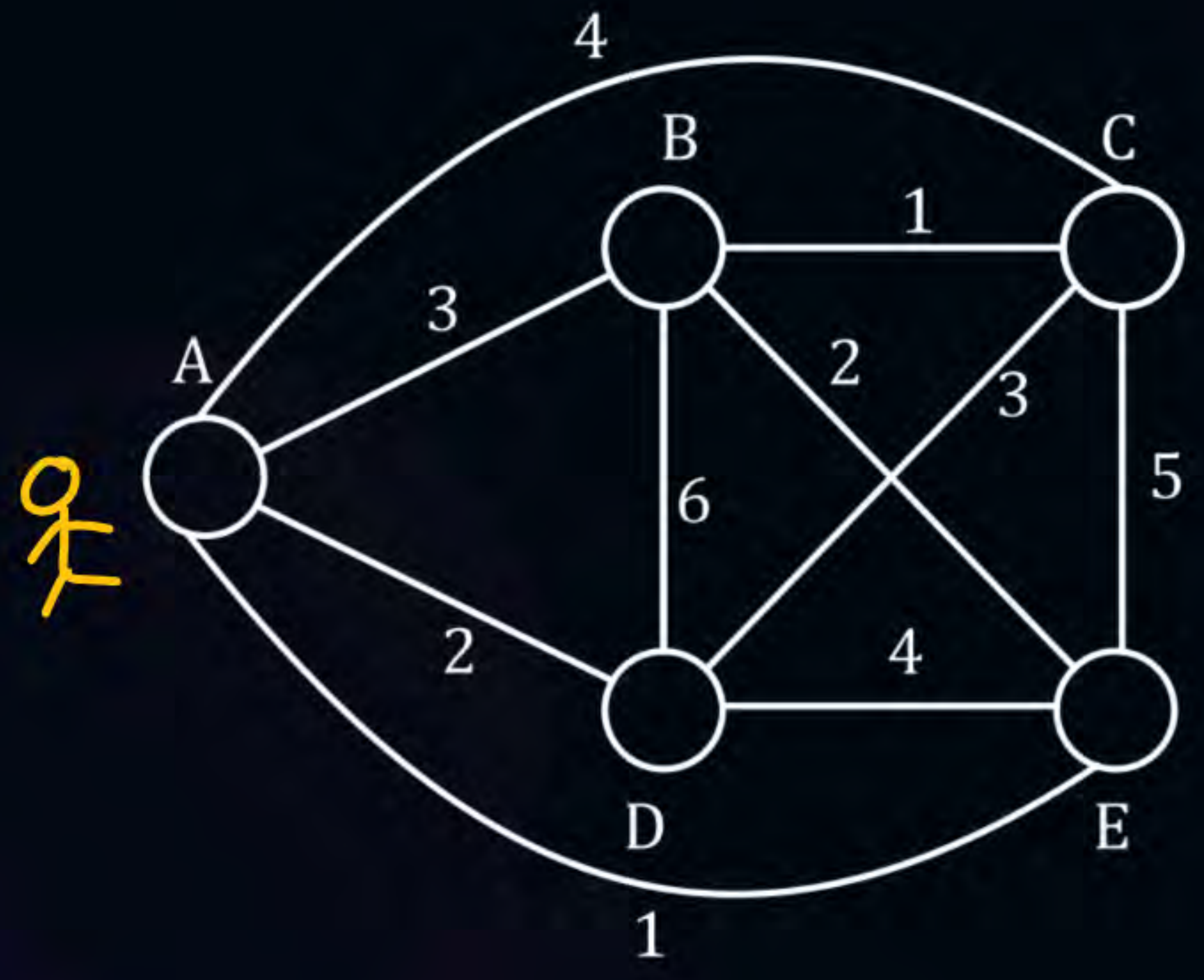
Multi stage graph problem

Topics to be Covered

Topic Travelling salesman problem

Topic Floyd Warshall

Topic: Travelling Salesman problem



A	B	C	D	E	A
A	C	B	D	E	A
A	D	B	C	E	A

$(n-1)!$

$O(n!) >> 2^n$

Topic: Travelling Salesman problem

- Detecting optimal sub structure:

Directed graph-

Cost matrix =
$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 4 & 2 \\ 1 & 2 & 0 & 5 \\ 3 & 4 & 1 & 0 \end{bmatrix} \end{array}$$

Topic: Travelling Salesman problem

$$T(1, \{2,3,4\}) = \min \begin{cases} (1, 2) + T(2, \{3, 4\}) \\ (1, 3) + T(3, \{2, 4\}) \\ (1, 4) + T(4, \{2, 3\}) \end{cases}$$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 4 & 2 \\ 3 & 1 & 2 & 0 & 5 \\ 4 & 3 & 4 & 1 & 0 \end{array}$$

Topic: Travelling Salesman problem

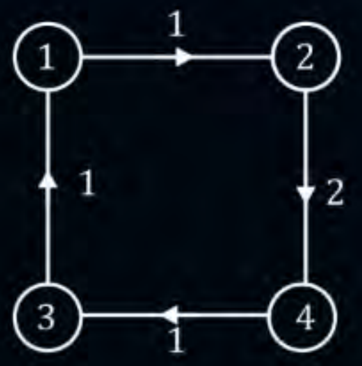
$$T(1, \{2,3,4\}) = \min \begin{cases} (1, 2) + T(2, \{3, 4\}) \\ (1, 3) + T(3, \{2, 4\}) \\ (1, 4) + T(4, \{2, 3\}) \end{cases}$$

$$T(2, \{3,4\}) = \min \begin{cases} (2, 3) + T(3, \{4\}) \\ (2, 4) + T(4, \{3\}) \end{cases}$$

$$T(3, \{2,4\}) = \min \begin{cases} (3, 2) + T(2, \{4\}) \\ (3, 4) + T(4, \{2\}) \end{cases}$$

$$T(4, \{2,3\}) = \min \begin{cases} (4, 2) + T(2, \{3\}) \\ (4, 3) + T(3, \{2\}) \end{cases}$$

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \left[\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 0 & 4 & 2 \\ 1 & 2 & 0 & 5 \\ 3 & 4 & 1 & 0 \end{array}\right] \end{array}$$



Topic: Travelling Salesman problem

$$T(1, \{2,3,4\}) = \min \begin{cases} (1, 2) + T(2, \{3, 4\}) \\ (1, 3) + T(3, \{2, 4\}) \\ (1, 4) + T(4, \{2, 3\}) \end{cases}$$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 4 & 2 \\ 3 & 1 & 2 & 0 & 5 \\ 4 & 3 & 4 & 1 & 0 \end{array}$$

$$T(2, \{3,4\}) = \min \begin{cases} (2, 3) + T(3, \{4\}) \\ (2, 4) + T(4, \{3\}) \end{cases}$$

$$T(3, \{2,4\}) = \min \begin{cases} (3, 2) + T(2, \{4\}) \\ (3, 4) + T(4, \{2\}) \end{cases}$$

$$T(4, \{2,3\}) = \min \begin{cases} (4, 2) + T(2, \{3\}) \\ (4, 3) + T(3, \{2\}) \end{cases}$$

$$T(3, \{4\}) = (3, 4) + T(4, \varnothing)$$
$$T(4, \{3\}) = (4, 3) + T(3, \varnothing)$$
$$T(2, \{4\}) = (2, 4) + T(4, \varnothing)$$

$$T(4, \{2\}) = (4, 2) + T(2, \varnothing)$$
$$T(2, \{3\}) = (2, 3) + T(3, \varnothing)$$
$$T(3, \{2\}) = (3, 2) + T(2, \varnothing)$$

Topic: Travelling Salesman problem

$$T(1, \{2,3,4\}) = \min \begin{cases} (1,2) + T(2, \{3,4\}) = 5 \\ (1,3) + T(3, \{2,4\}) = 9 \\ (1,4) + T(4, \{2,3\}) = 7 \end{cases}$$

$$T(2, \{3,4\}) = \min \begin{cases} (2,3) + T(3, \{4\}) = 12 \\ (2,4) + T(4, \{3\}) = 4 \end{cases}$$

$$T(3, \{2,4\}) = \min \begin{cases} (3,2) + T(2, \{4\}) = 7 \\ (3,4) + T(4, \{2\}) = 10 \end{cases}$$

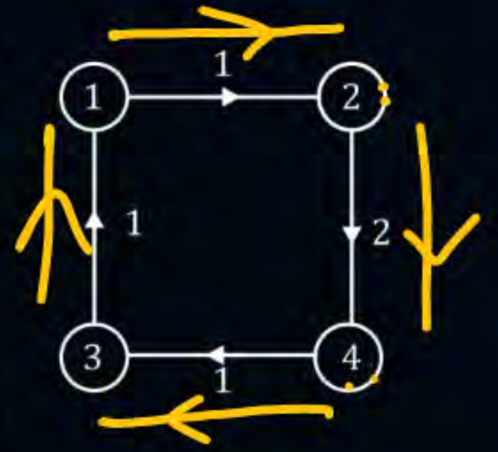
$$T(4, \{2,3\}) = \min \begin{cases} (4,2) + T(2, \{3\}) = 9 \\ (4,3) + T(3, \{2\}) = 4 \end{cases}$$

$T(3, \{4\}) = (3, 4) + T(4, \varnothing) = 8$

$T(4, \{3\}) = (4, 3) + T(3, \varnothing) = 2$

$T(2, \{4\}) = (2, 4) + T(4, \varnothing) = 5$

	1	2	3	4
1	0	1	2	3
2	1	0	4	2
3	1	2	0	5
4	3	4	1	0



$T(4, \{2\}) = (4, 2) + T(2, \varnothing) = 5$

$T(2, \{3\}) = (2, 3) + T(3, \varnothing) = 5$

$T(3, \{2\}) = (3, 2) + T(2, \varnothing) = 3$

$T(2, \varnothing) = (2, 1) = 1$ $\quad T(3, \varnothing) = (3, 1) = 1$

$T(4, \varnothing) = (4, 1) = 3$

Minimum is 5; To travel from 1 to {2,3,4} and come back 1 - minimum cost is 5.

Topic: Travelling Salesman problem

Bottom up dynamic programming algorithm

Recursive equation:

$$T(i, s) = \min_{j \in S}((i, j) + T(j, S - \{j\})) \; ; \; S \neq \emptyset$$

$$= (i, 1) \; ; \; S = \emptyset$$

Topic: Travelling Salesman problem

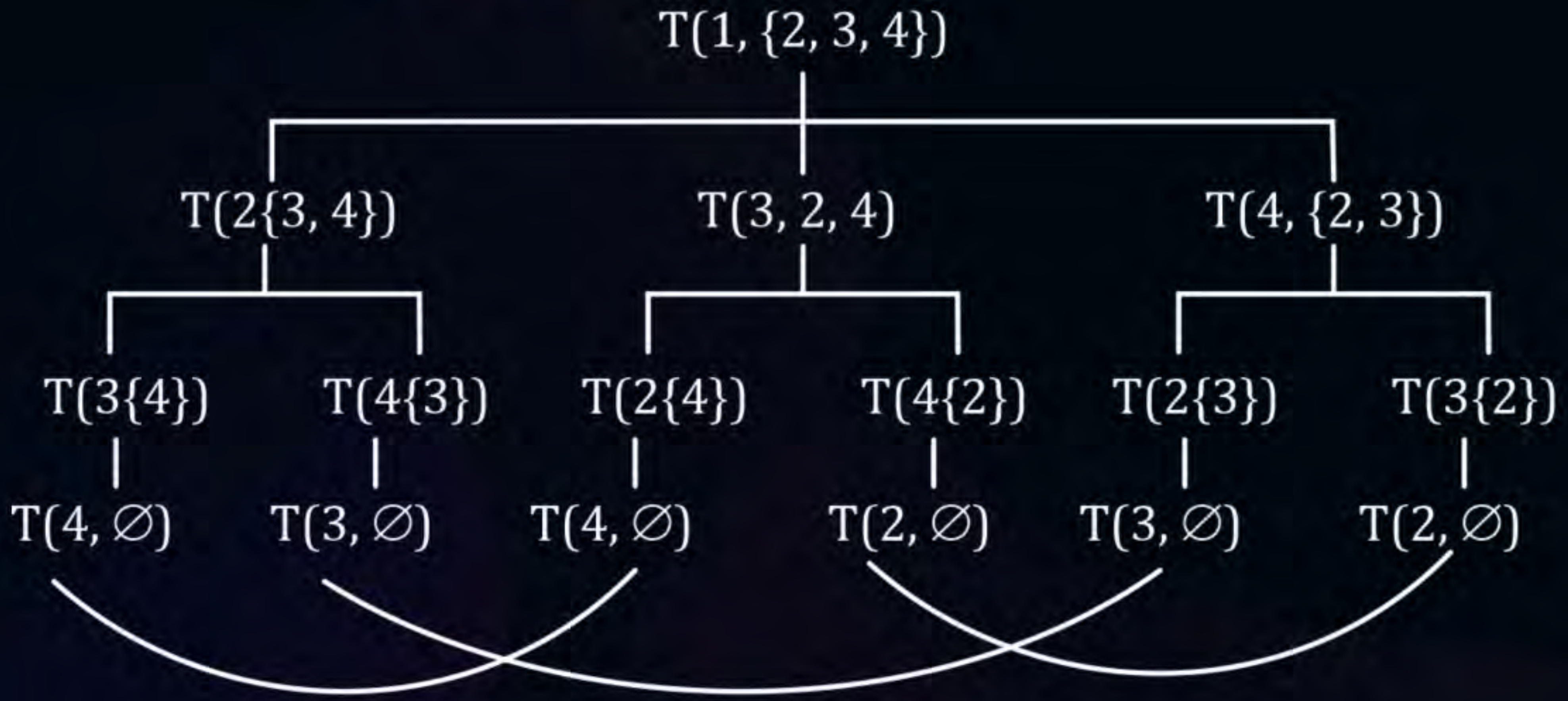
Bottom up dynamic programming algorithm

Recursive equation:

$$T(i, s) = \min_{j \in S}((i, j) + T(j, S - \{j\})) \; ; S \neq \emptyset$$

$$= (i, 1) \; ; S = \emptyset$$

$V = \{1,2,3,4\}$



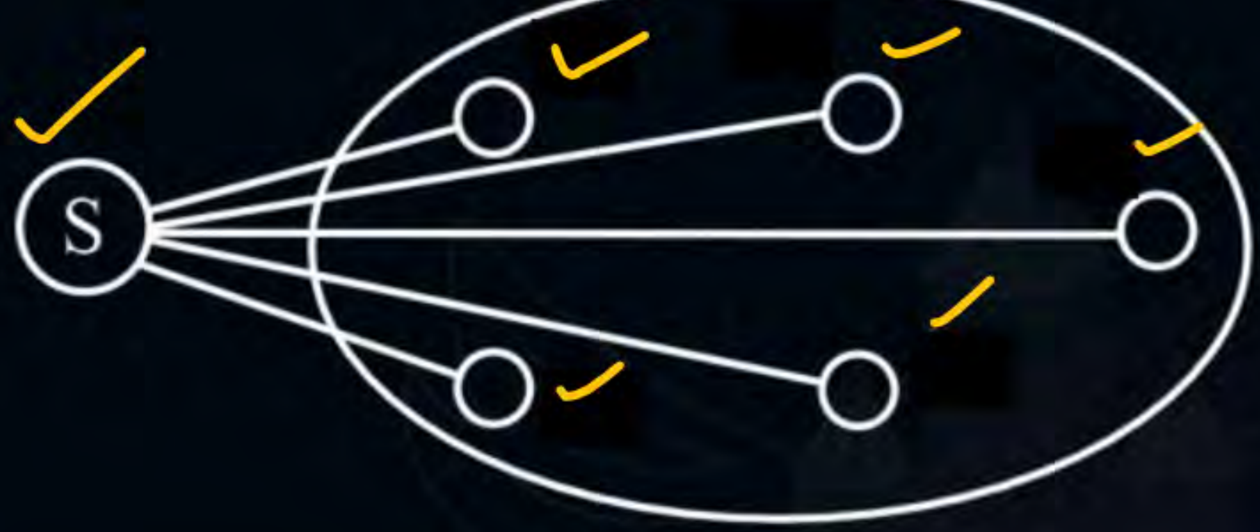
Time complexity $= O(n\, 2^n) << O(n!)$

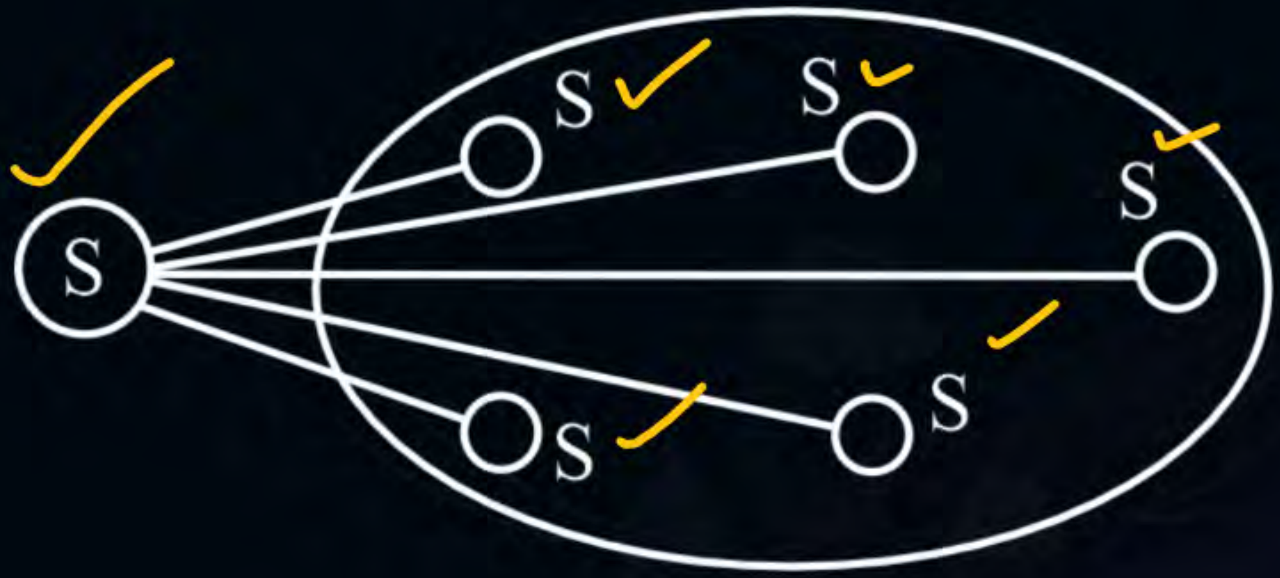
Space complexity $= O(n^2\, 2^n)$ ✓

$(n 2^n)$ ✓

Topic: All Pairs Shortest Path – Floyd Warshall

- Relationship between single source shortest path and all pairs shortest path–





Topic: All Pairs Shortest Path – Floyd Warshall

- You could run single source shortest path from every vertex then will become all pairs shortest path.

To implement SSSP (Time)	All pair shortest path (time)
Dijkstra Algo – O (E log V)	O (V.E log V) = O (V . V^2 log V) = O (V^3 log V)
Bellman ford Algo – O (V E)	O (V^2 E) = O(V^2 . V^2) = O (V^4)

Topic: All Pairs Shortest Path – Floyd Warshall

$V = \{1,2,3,4\}$

And $i, j \in V$

$$D_{ij}^{k} = \text{“path”} = \min\begin{cases} D_{ij}^{(k-1)} \\ D_{ik}^{(k-1)} + D_{kj}^{(k-1)} \end{cases}$$

Topic: Optimal Substructure

Example:

$$D^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 11 & 1 & 6 \\ 11 & 0 & 7 & 3 \\ 1 & 7 & 0 & 2 \\ 6 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

Topic: Optimal Substructure

Example:

$$D^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 11 & 1 & 6 \\ 11 & 0 & 7 & 3 \\ 1 & 7 & 0 & 2 \\ 6 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

$$D^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 11 & 1 & 6 \\ 11 & 0 & 7 & 3 \\ 1 & \cancel{7} & 0 & 2 \\ 6 & \cancel{3} & \cancel{2} & 0 \end{bmatrix} \end{array}$$

Distance of the path in which might use 1 or might not use 1.

$(2, 3) = \min(7, 2 \to 1 \to 3) = 7$

$(2, 4) = \min(3, 2 \to 1 \to 4) = 3$

$(3,4) = \min(2, \underbrace{3 \to 1 \to 4}_{1+6}) = 2$

Topic: Optimal Substructure

Example:

$$D^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & \cancel{1} & 7 & 0 & 2 \\ 4 & \cancel{6} & 3 & \cancel{2} & 0 \end{array}$$

$$(1, 3) = (1, \overset{11}{1 \to 2} + 2 \to 3) = 1$$

$$(1, 4) = (6, \underset{11}{1 \to 2} + 2 \to 4) = 6$$

$$(3, 4) = (2, \underset{7}{3 \to 2} + 2 \to 4) = 2$$

Topic: Optimal Substructure

Example:

$$D^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 1 & 3 \\ 2 & \cancel{8} & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & \cancel{3} & \cancel{3} & 2 & 0 \end{array}$$

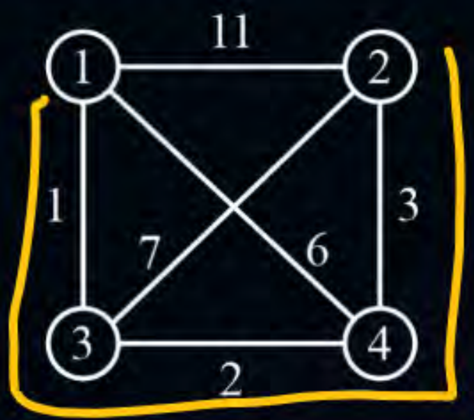
$(1, 2) = (11, \overset{1}{1 \to 3} + \overset{7}{3 \to 2}) = 8$ ✓

$(1, 4) = (6, \overset{1}{1 \to 3} + \overset{2}{3 \to 4}) = 3$ ✓

$(2, 4) = (3, \underbrace{2 \to 3}_{7} + 3 \to 4) = 3$

Topic: Optimal Substructure

Example:



$$D^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$$D^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 1 & 3 \\ 2 & 8 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 3 & 3 & 2 & 0 \end{array}$$

$$D^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 6 & 1 & 3 \\ 2 & 6 & 0 & 5 & 3 \\ 3 & 1 & 5 & 0 & 2 \\ 4 & 3 & 3 & 2 & 0 \end{array}$$ ✓

$(1, 2) = (8, \overset{3}{\underline{1 \to 4}} + \overset{3}{\underline{4 \to 2}}) = \underline{6}$ ✓

$(1, 3) = (1, \underline{1 \to 4 + 4 \to 3}) = 1$

$(2, 3) = (7, \underset{3}{\underline{2 \to 4}} + \underset{2}{\underline{4 \to 3}}) = \underline{5}$

Topic: Optimal Substructure

Subproblem:

$= n$ matrices/vertices $\times$ size of each one is n^2

$= n\,(n^2)$

$= O(n^3)$ – no. of problems

Total time complexity $= O(n^3)$

Space complexity $= O(n^2)$ ---at any point of time only 2 matrices are used

Topic: Optimal Substructure

Algorithm:

FLOYD_WARSHALL (w) {

(1) n = w . rows // vertices

(2) $D^0 = w$

(3) for k = 1 to n // vertices

(4) Let $D^{(k)} = (d_{ij}^{k})$ be a n × n matrix

(5) for i = 1 to n

(6) for j = 1 to n

(7) $$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

(8) return $D^{(n)}$

}

$T(n) = O(n^3)$

$S(n) = O(n^2) \rightarrow D^0, D^1, D^2, D^3, \ldots, D^n$

THANK - YOU