17/10/19

Module - IV

Dynamic Programming

Greedy and dynamic programming are optimization problem, ie, the result should be either minimum/maximum. Dynamic programming is an algorithm designing method that can be used when a sol to the problem may be viewed as sequence of decision.

Dynamic programming is a method for efficiently solving a broad range of search optimization problems. DP computes all feasible solution and the picks the best one as feasible solution. Disadvantage of DP is time consuming.

eg: fibonacci

```
fib(int n)
{
  if (n ≤ 1)
    return 0 and 1;
  else
    return fib(n-1) + fib(n-2);
}
```

Recursive relation

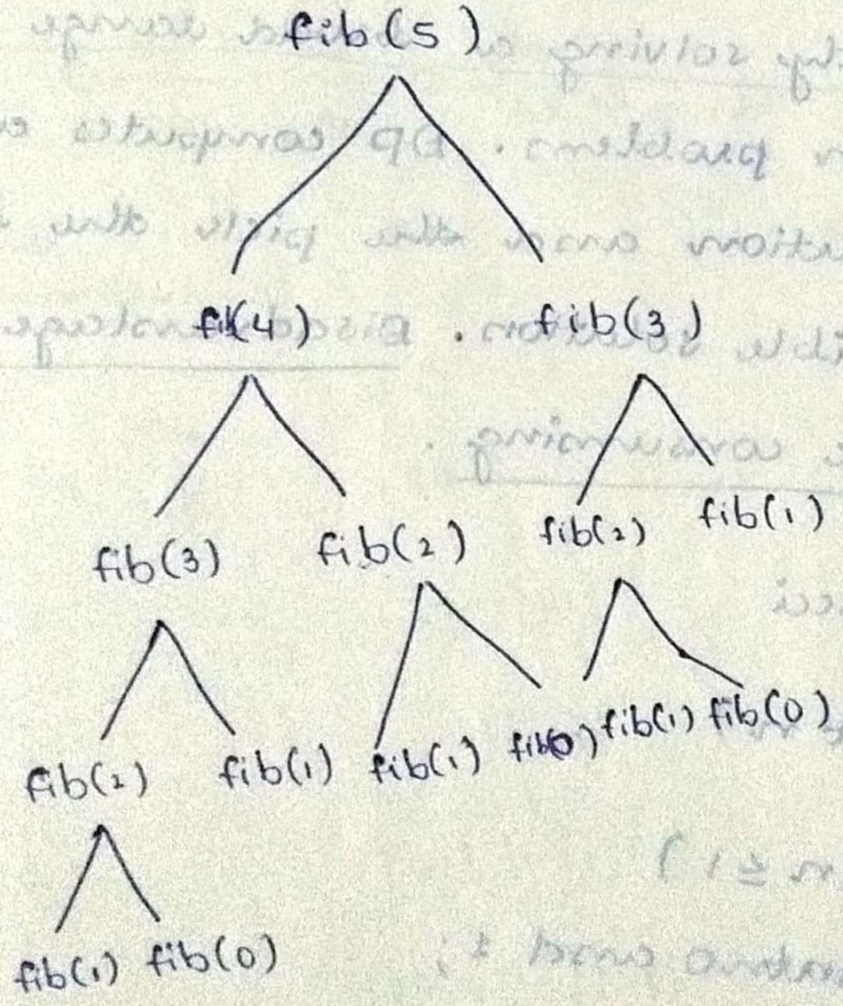
$$f(n) = \begin{cases} 0 & , n = 0 \\ 1 & , n = 1 \\ n-2+n-1 & , n \geq 1 \end{cases}$$

Time complexity,

$$f(n) = 2T(n-1) + 1$$

Using Master's theorem,

$$f(n) = O(2^n)$$



Inorder to find fib(5), we need to divide the problem into two recursive function fib(4) and fib(3). This sub division continuous until we reach fib(0) and fib(1) in the leaf node. This causes duplication. This is called overlapping subproblem

- also called divide and conquer approach

D & C leads to several overlapping instances inorder to solve this kind of overlapping we use a programming structure called dynamic programming.

The development of DP algorithm can be broken into a sequence of 4 steps.

step 1 : characterise the structure of an optimal sol.

2 : Recursively design the value of and optimum sol.

3 : compute the value in bottom up fashion.

Step 4: construct an optimum system from the computed information.

Optimal sub-structure & principle of optimality

A problem is said to have an optimal sub-systems, if the globally optimized sys. can be constructed from locally constructed systems. The principle of optimality says that an optimal sequence of division have the property that what ever the initial state and decision are, the remaining decision must constitute an optimal decision sequence with regard to the state resulting from the first decision.

The time complexity of divide and conquer approach / recursive call is very high (eg: $O(2^n)$ - fib). The same func calls again and again in D&C. one of the method in DP is Memonization

In Memonization, we do not call

all fns., recursively, we call the info. within an array.

0	1	2	3	4	5
0	1	1			
f(0)	f(1)	f(2)			

So, the time complexity is: O(n+1).

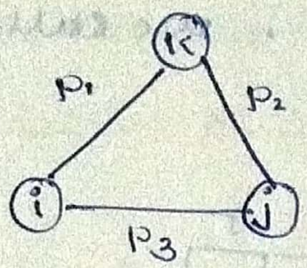
Here, 1 is constant and time complexity is O(n).

∴ The exponential time complexity using D&C can be changed to linear using DP. ie, O(n).

18/10/19

All pair shortest path problem

This problem is to find the shortest path btw all vertices in a given graph. we start this problem by writing the cost or distance, using cost adjacency matrix. Then we refine all the vertex by replacing the given cost by a minimum cost.



Floyd warshal's algorithm, consider all intermediate vertex of a shortest path, if k is an intermediate vertex of path B, then we decompose p into $i \xrightarrow{p_1} k$ and $k \xrightarrow{p_2} j$. If k is not an intermediate vertex of path B, then all intermediate vertex of path B are intersect.

Recursive Function

$$d_{ij} = \begin{cases} w_{ij}, & \text{if } k = 0 \\ \min\left[d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right], & \text{if } k \geq 1 \end{cases}$$

Shortest path (cost, A, n)

// cost [i, j] is the cost adjacency matrix

A[i, j] is the shortest path from i to j

cost [i, j] = 0 , if i = j

{

//if there is no intermediate node

for i = 1 to n do

$A[i, j] = cost[i, j]$

//if there is an intermediate node

for k = 1 to n

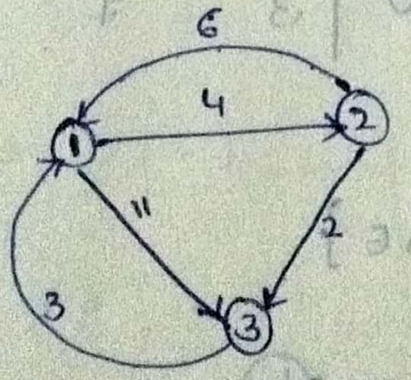
for i = 1 to n

for j = 1 to n

$A[i, j] = \min\left[A[i,j],\ A[i, k] + A[k, j]\right]$

}

* Find the minimum cost for the following graph using All pair Shortest Path Problems.



cost adjacency matrix:

	1	2	3
1	0	4	11
2	6	0	2
3	∞	∞	0

$A_0 = \{0\}$

$A_1 \overset{(via)}{=} \{0, 1\}$

	1	2	3
1	0	4	11
2	6	0	2
3	3	7	0

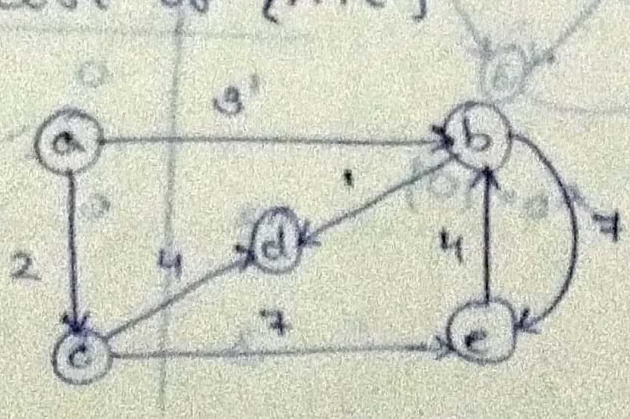
$A_2 = \{0, 1, 2\}$

	1	2	3
1	0	4	6
2	6	0	2
3	3	7	0

$A_3 = \{0, 1, 2, 3\}$

	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0

* Find cost of $\{A, E\}$



$A_0 =$ a

	a	b	c	d	e
a	0	3	2	∞	∞
b	∞	0	∞	1	7
c	∞	∞	0	4	7
d	∞	∞	∞	0	∞
e	∞	4	∞	∞	0

$$A_1 = \{\overset{a}{0}, \overset{b}{1}\}$$

	a	b	c	d	e
a	0	3	2	4	10
b	∞	0	∞	1	7
c	∞	∞	0	4	7
d	∞	∞	∞	0	∞
e	∞	4	∞	5	0

$A_2 = \{a, b, c\}$

	a	b	c	d	e
a	0	3	2	4	9
b	∞	0	∞	1	7
c	∞	∞	0	4	7
d	∞	∞	∞	0	∞
e	∞	4	∞	5	0

$A_3 = \{a, b, c, d\}$

	a	b	c	d	e
a	0	3	2	4	9
b	∞	0	∞	1	7
c	∞	∞	0	4	7
d	∞	∞	∞	0	∞
e	∞	4	∞	5	0

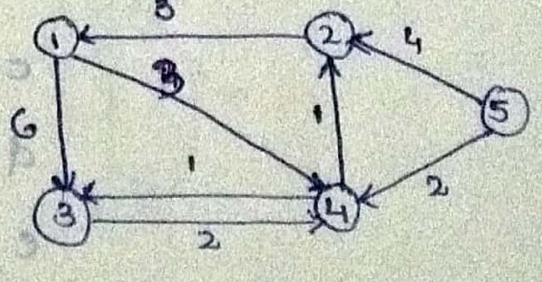
$A_4 = \{a, b, c, d, e\}$

	a	b	c	d	e
a	0	3	2	4	9
b	∞	0	∞	1	7
c	∞	11	0	4	7
d	∞	∞	∞	0	∞
é	∞	4	∞	5	0

$(A - E) = 9$

21/10/19

* Find All pair shortest path Problem for the following graph.



$A_0 = \{0\}$

	1	2	3	4	5
1	0	α	6	3	α
2	3	0	α	α	α
3	α	α	0	2	α
4	α	1	1	0	α
5	α	4	α	2	0

$A_1 = \{0, 1\}$

	1	2	3	4	5
1	0	α	6	3	α
2	3	0	9	6	α
3	α	α	0	2	α
4	α	1	1	0	α
5	α	4	α	2	0

$A_2 = \{0, 1, 2\}$

	1	2	3	4	5
1	0	α	6	3	α
2	3	0	9	6	α
3	α	α	0	2	α
4	4	1	1	0	α
5	7	4	13	2	0

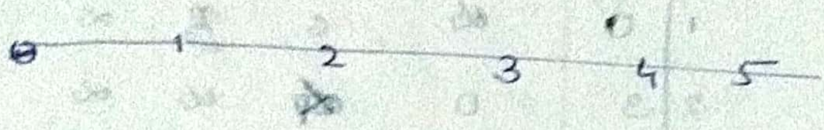
$A_3 = \{0; 1, 2, 3\}$

	1	2	3	4	5
1	0	α	6	3	α
2	3	0	9	6	α
3	[illegible]	[illegible]	0	2	α
4	4	1	1	0	α
5	7	4	13	2	0

$A_4 = \{0, 1, 2, 3, 4\}$

	1	2	3	4	5
1	0	4	4	3	α
2	3	0	7	6	α
3	6	3	0	2	α
4	4	1	1	0	α
5	6	3	3	2	0

~~$A_0 = \{0, 1, 2, 3, 4, 5\}$~~



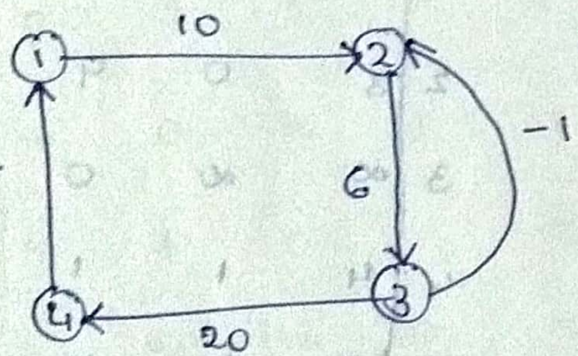
Path and cost =

$5 \to 1 : 5 - 4 - 2 - 1 = 6$

Path = $\{5, 4, 2, 1\}$

Cost = $\underline{6}$

*



$A_0 = \{0\}$

	1	2	3	4
1	0	10	α	α
2	α	0	6	α
3	α	-1	0	20
4	-5	α	α	0

$A_1 = \{0, 1\}$

	1	2	3	4
1	0	10	α	α
2	α	0	6	α
3	α	-1	0	20
4	-5	5	α	0

$A_2 = \{0, 1, 2\}$

	1	2	3	4
1	0	10	16	∞
2	∞	0	6	∞
3	4	-1	0	20
4	-5	5	11	0

$A_3 = \{0, 1, 2, 3\}$

	1	2	3	4
1	0	10	16	36
2	21	0	6	26
3	15 ~~∞~~	-1	0	20
4	-5	5	11	0

cost $\{1-4\} = 36$.

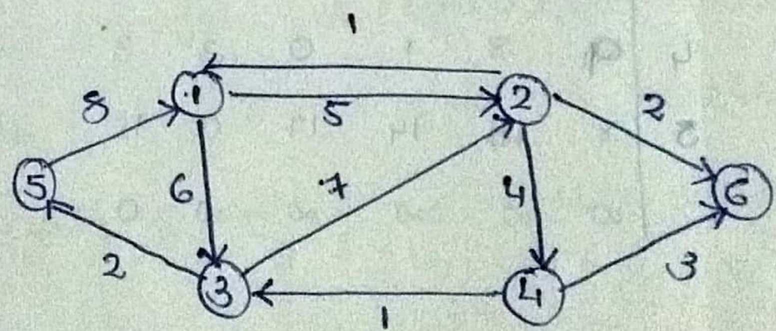
path $= \{1, 2, 3, 4\}$

---

* Find the minimum cost using All pair shortest path Problem for the following graph.

(i) 1-6

(ii) 4-1



$A_0 =$

	1	2	3	4	5	6
1	0	5	6	∞	∞	∞
2	1	0	∞	4	∞	2
3	∞	7	0	∞	2	∞
4	∞	∞	1	0	∞	3
5	8	∞	∞	∞	0	∞
6	∞	∞	∞	∞	∞	0

$A_1 = \{0, 1\}$

	1	2	3	4	5	6
1	0	5	6	∞	∞	∞
2	1	0	7	4	∞	2
3	∞	7	0	∞	2	∞
4	∞	∞	1	0	∞	3
5	8	13	14	∞	0	∞
6	∞	∞	∞	∞	∞	0

$A_2 = \{0, 1, 2\}$

	1	2	3	4	5	6
1	0	5	6	9	∞	7
2	1	0	7	4	∞	2
3	8	7	0	11	2	9
4	∞	∞	1	0	∞	3
5	8	13	14	17	0	15
6	∞	∞	∞	∞	∞	0

$A_3 = \{0, 1, 2, 3\}$

	1	2	3	4	5	6
1	0	5	6	9	8	7
2	1	0	7	4	9	2
3	8	7	0	11	2	9
4	9	8	1	0	3	3
5	8	13	14	17	0	15
6	∞	∞	∞	∞	∞	0

$A_4 = \{0, 1, 2, 3, 4\}$

	1	2	3	4	5	6
1	0	5	6	9	8	7
2	1	0	5	4	7	2
3	8	7	0	11	2	9
4	9	8	1	0	3	3
5	8	13	14	17	0	15
6	∞	∞	∞	∞	∞	0

$A_5 = \{0, 1, 2, 3, 4, 5\}$

	1	2	3	4	5	6
1	0	5	6	9	8	7
2	1	0	5	4	7	2
3	8	7	0	11	2	9
4	9	8	1	0	3	3
5	8	13	14	17	0	15
6	∞	∞	∞	∞	∞	0

$\Rightarrow$ cost (1-6) = 7

path = {1, 2, 6}

$\Rightarrow$ cost (4-1) = 9

path = {4, 3, 2, 1}

24/10/19

Travelling Sales Person Problems

Given a graph $G = \{V, E\}$. C_{ij} is the edge having cost C_{ij}, a path $V_i \longrightarrow V_j$ are adjacent vertex. Then starts from your city and move through various intermediate city atleast once and reach the source or return back to starting vertex. A tour of G is directed single cycle that include every vertex in V. ~~cost~~

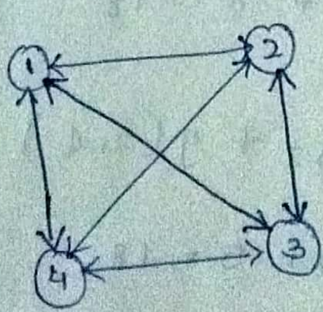
cost = sum of weight of edges on a tour.

Then the principle of optimality,

$$g(i,s) = \min_{k \in S} \left(C_{ik} + g(k, \{S-k\}) \right)$$

Let $g(i,s)$ is the length of shortest path from node i and going through all vertices in S and terminating at the source vertex at 1.

eg:



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & & & & \\ 2 & & \left[\begin{matrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{matrix}\right] & & \\ 3 & & & & \\ 4 & & & & \end{array}$$

If intermediate node $|S| = 0$, starting from 1 and ending at 1 without any intermediate vertex.

$V_S = 1 = V_T$

$g(2, \phi) = C_{21} = 5$ // starting @ 2 & ending @ 1, no intermediate

$g(3, \phi) = C_{31} = 6$

$g(4, \phi) = C_{41} = 8$

Find out a path from starting vertex to destination via a single vertex, ie.

$|S| = 1$.

$g(2, \{3\})$, starting from 2 to 1 via 3 (2-3-1)

$g(2, \{3\}) = C_{23} + g(3, \phi)$ // (2-3-1)

$= 9 + 6 = 15$

$g(2, \{4\}) = C_{24} + g(4, \phi)$ // (2-4-1)

$= 10 + 8 = 18$

$g(3, \{2\}) = C_{32} + g(2, \phi)$ // 3-2-1

$= 13 + 5 = 18$

$g(3, \{4\}) = C_{34} + g(4, \phi)$ // 3-4-1

$= 12 + 8 = 20$

$g(4, \{3\}) = c_{43} + g(3, \phi)$ // 4-3-1

$= 9 + 6 = 15$

$g(4, \{2\}) = c_{42} + g(2, \phi)$ // 4-2-1

$= 8 + 5 = 13$

If $|S| = 2$, ie. consider two intermediate vertex.

$g(2, \{3,4\}) \Rightarrow$ starting from 2 to 1 via 3 & 4 (2-3-4-1)

$g(2, \{3,4\}) = \min\{c_{23} + g(3, \{4\}),\ c_{24} + g(4, \{3\})\}$

$= \min\{9 + 20,\ 10 + 15\}$

$= \min\{29, 25\} = 25$

$g(3, \{2,4\})$ from 3rd vertex to 1 via 2 intermediate vertex 2 & 4   3-2-4-1

$g(3, \{2,4\}) = \min(c_{32} + g(2, \{4\});\ c_{34} + g(4, \{2\})$

$= \min(13 + 18,\ 12 + 13)$

$= \min(31, 25) = 25$

$g(4, \{2,3\} = \min(c_{42}, g(2, \{3\}),\ c_{43} + g(3, \{2\}))$

4-2-3-1 $= \min(8+15,\ 9+18) \Rightarrow \min(23, 27) = 23$

If $|S| = 3$ intermediate vertex, ie, $g(1, \{2,3,4\})$

=> starting from 1 ending at 1 via 2-3-4

$$g(1, \{2,3,4\}) = \min \begin{cases} c_{12} + g(2, \{3,4\}), \\ c_{13} + g(3, \{2,4\}), \\ c_{14} + g(4, \{2,3\}) \end{cases}$$

$$= \min \begin{cases} 10 + 25, \quad 15 + 25, \\ 20 + 23 \end{cases}$$

$$= \min \{35, 40, 43\}$$

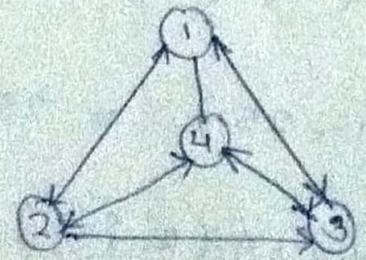
$$= 35$$

Optimal path $= c_{12} + g(2, \{3,4\})$

$= 1 - 2 - 4 - 3 - 1$

25/10/19

* Find the shortest path using of travelling sales person using dynamic programming.



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 10 & 15 & 20 \\ 10 & 0 & 35 & 25 \\ 15 & 35 & 0 & 30 \\ 20 & 25 & 30 & 0 \end{bmatrix} \end{array}$$

$|S| = 0$

$V_S = 1 = V_T$

$g(2, \phi) = C_{21} = 10$ $\quad g(4, \phi) = C_{41} = 20$

$g(3, \phi) = C_{31} = 15$

$|S| = 1$

$g(2, \{3\}) = C_{23} + g(3, \phi) = 35 + 15 = 50$

$g(2, \{4\}) = C_{24} + g(4, \phi) = 25 + 20 = 45$

$g(3, \{2\}) = C_{32} + g(2, \phi) = 35 + 10 = 45$

$g(3, \{4\}) = C_{34} + g(4, \phi) = 30 + 20 = 50$

$g(4, \{2\}) = C_{42} + g(2, \phi) = 25 + 10 = 35$

$g(4, \{3\}) = C_{43} + g(3, \phi) = 30 + 15 = 45$

$|S| = 2$

$g(2, \{3, 4\}) = \min\{C_{23} + g(3, \{4\}), C_{24} + g(4, \{3\})\}$

$= \min\{35 + 50, 25 + 45\} = \min\{85, 70\}$

$= 70$

$$g(3, \{2,4\}) = \min \begin{cases} c_{32} + g(2, \{4\}), \\ c_{34} + g(4, \{2\}) \end{cases}$$

$$= \min \{35 + 45, 30 + 35\}$$

$$= \min \{80, 65\}$$

$$= 65$$

$$g(4, \{2,3\}) = \min \begin{cases} c_{42} + g(2, \{3\}), \\ c_{43} + g(4, \{2\}) \end{cases}$$

$$= \min \{25 + 50, 30 + 35\}$$

$$= \min \{75, 65\}$$

$$= 65$$

$|S| = 3$

$$g(1, \{2,3,4\}) = \min \begin{cases} c_{12} + g(2, \{3,4\}) \\ c_{13} + g(3, \{2,4\}) \\ c_{14} + g(4, \{2,3\}) \end{cases}$$

$$= \min \{10 + 70, 15 + 65, 20 + 65\}$$

$$= \min \{80, 80, 85\}$$

Scanned with CamScanner

optimal path $= \{ c_{12} + g(2, \{3,4\}) \}$

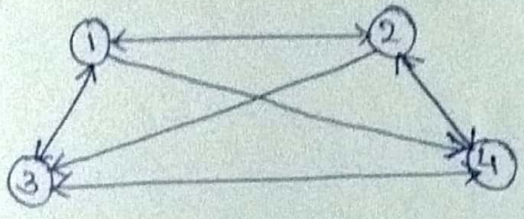
$= \{ c_{13} + g(3, \{2,4\}) \}$

$= 1-2-4-3-1$

$= 1-3-4-2-1$

---

*



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 25 & 12 \\ 2 & 5 & 0 & 10 & 20 \\ 3 & 25 & 10 & 0 & 9 \\ 4 & 12 & 20 & 9 & 0 \end{array}$$

$|S| = 0$

$g(2, \phi) = c_{21} = 5$

$g(3, \phi) = c_{31} = 25$

$g(4, \phi) = c_{41} = 12$.

$|S| = 1$

$g(2, \{3\}) = c_{23} + g(3, \phi) = 10 + 25 = 35$

$g(2, \{4\}) = c_{24} + g(4, \phi) = 20 + 12 = 32$

$g(3, \{4\}) = c_{34} + g(4, \phi) = 9 + 12 = 21$

$g(3, \{2\}) = c_{32} + g(2, \phi) = 10 + 5 = 15$

$g(4,\{2\}) = c_{42} + g(2,\phi) = 20 + 5 = 25$

$g(4,\{3\}) = c_{43} + g(3,\phi) = 9 + 25 = 34.$

$|S| = 2.$

$g(2,\{4,3\}) = \{c_{24} + g(4,\{3\}),\ c_{23} + g(3,\{4\})$

$= \{20 + 34,\ 10 + 21$

$= \{54, 31\}$

$= 31$ //

$g(3,\{4,2\}) = \{c_{34} + g(4,\{2\}),\ c_{32} + g(2,\{4\})$

$= \{9 + 25,\ 10 + 32$

$= \{34, 42\}$

$= 34$ //

$g(4,\{3,2\}) = \{c_{43} + g(3,\{2\}),\ c_{42} + g(2,\{3\})$

$= 9 + 15,\ 20 + 25$

$= 24, 45 = 24$ //

$|S| = 3$

$g(1,\{2,3,4\}) = c_{12} + g(2,\{3,4\}),\ c_{13} + g(3,\{2,4\}),$

$c_{14} + g(4,\{2,3\})$

$= \{5 + 31,\ 25 + 34,\ 12 + 24$

$= \{36, 59, 36\}$

min $= 36$.

optimal path $= \{ c_{12} + g(2, \{3, 4\})$

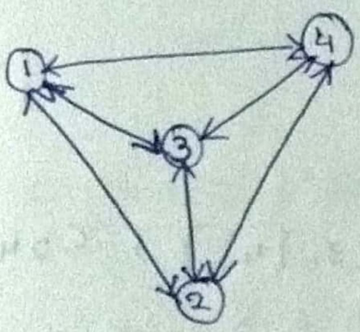
$= 1 - 2 - 3 - 4 - 1$

$= \{ c_{14} + g(4, \{2, 3\})$

$= 1 - 4 - 3 - 2 - 1$

softcopy

*



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 2 & 10 & 3 \\ 2 & 6 & 0 & 1 & 8 \\ 3 & 2 & 3 & 0 & 5 \\ 4 & 9 & 7 & 4 & 0 \end{array}$$

$|s| = 0$

$g(2, \phi) = c_{21} = 6$

$g(3, \phi) = c_{31} = 2$

$g(4, \phi) = c_{41} = 9$

$|s| = 2$

$g(2, \{3\}) = c_{23} + g(3, \phi) = 1 + 2 = 3$

$$g(2,\{4\}) = C_{24} + g(4,\phi)$$

$$= 8 + 9 = 17$$

$$g(3,\{2\}) = C_{32} + g(2,\phi)$$

$$= \overset{3}{\cancel{10}} + 6 = 9$$

$$g(3,\{4\}) = C_{34} + g(4,\phi)$$

$$= 5 + 9 = \overset{14}{\cancel{7}}$$

$$g(4,\{3\}) = C_{43} + g(3,\phi)$$

$$= 4 + 2 = 6$$

$$g(4,\{2\}) = C_{42} + g(2,\phi)$$

$$= \overset{7}{\cancel{20}} + 6 = \overset{13}{\cancel{26}}$$

$|S| = 2$

$$g(2,\{3,4\}) = \min\{ c_{23} + g(3,\{4\}),\ c_{24} + g(4,\{3\})$$

$$= \min\{ \overset{1}{\cancel{10}} + \overset{14}{\cancel{7}},\ 8 + 6$$

$$= \min\{ \overset{15}{\cancel{17}},\ 14\} = \cancel{6}\ 14$$

$$g(3,\{2,4\}) = \min\{ c_{32} + g(2,\{4\}),\ c_{34} + g(4,\{2\})$$

$$= \min\{ 3 + \overset{17}{\cancel{9}},\ 5 + 13$$

$$= \min\{20,\ 18\}$$

$$= 18$$

$g(4, \{2, \{3\}) = \{c_{42} + g(2, \{3\}), \; c_{43} + g(3, \{2\})\}$

$= \{\cancel{20}7 + 3, \; 4 + 9\} = \{10, 13\} = 10$

$|S| = 3$

$$g(1, \{2,3,4\}) = \min \{ c_{12} + g(2, \{3,4\}), * \quad c_{13} + g(3, \{2,4\}), * \quad c_{14} + g(4, \{2,3\}) \}$$

$= \min \{2 + 14, \; 10 + 18, \; 3 + 10$

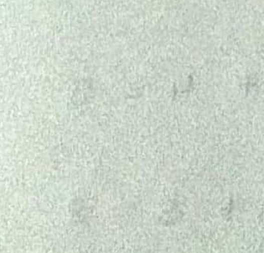
$= \min \{16, 28, 13\}$

$\min = 13$

optimal path $= c_{14} + g(4, \{2,3\})$

$= 1 - 4 - 2 - 3 - 1$

*



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 5 & 3 \\ 2 & 4 & 0 & 3 & 5 \\ 3 & 2 & 1 & 0 & 3 \\ 4 & 5 & 7 & 2 & 0 \end{array}$$

$|S| = 0$

$V_S = 1 = V_T$

$g(2, \phi) = C_{21} = 4$

$g(3, \phi) = C_{31} = 2$

$g(4, \phi) = C_{41} = 5$

$|S| = 1$

$g(2, \{3\}) = C_{23} + g(3, \phi) = 3 + 2 = 5$

$g(2, \{4\}) = C_{24} + g(4, \phi) = 5 + 5 = 10$

$g(3, \{2\}) = C_{32} + g(2, \phi) = 1 + 4 = 5$

$g(3, \{4\}) = C_{34} + g(4, \phi) = 3 + 5 = 8$

$g(4, \{2\}) = C_{42} + g(2, \phi) = 7 + 4 = 11$

$g(4, \{3\}) = C_{43} + g(3, \phi) = 2 + 2 = 4$

$|S| = 2$

$g(2, \{3,4\}) = \{ C_{23} + g(3, \{4\}), \quad C_{24} + g(4, \{3\})$

$= \min\{3+8, 5+4\} = \{11, 9\} = 9$

$g(3, \{2,4\}) = \min\{c_{32} + g(2,\{4\}), c_{34} + g(4,\{2\})$

$= \min\{1+10, 3+11$

$= \min\{11, 14\} = 11$

$g(4, \{2, \{3\}) = \min\{c_{42} + g(2, \{3\}), c_{43} + g\{3, \{2\}$

$= \min\{7+5, 2+5$

$= \min\{12, 7\} = 7$

$|s| = 3$

$$g(1, \{2,3,4\}) = \min \begin{cases} c_{12} + g(2, \{3,4\}), \\ c_{13} + g(3, \{2,4\}), \\ c_{14} + g(4, \{2,3\}). \end{cases}$$

$= \min\{8+9, 5+11, 3+7$

$= \min\{17, 16, 10\} = 10$

min $= 10$

optimal path $= c_{14} + g(4, \{2,3\})$

$= 1-4-3-2-1$

? Divide and Conquer Vs Dynamic Programming

Divide and Conquer	Dynamic Programming.
→ It involves three steps at each level of recursion: · Divide the problems into a no. of subproblems. · conquer the subproblems by solving them recursively. · combine the sol. to the subproblems into the sol. for original subproblems.	→ It involves the sequence of four steps: · Characterize the structure of optimal solutions. · Recursively defines the values of optimal sols. · Compute the value of optimal sols. in a Bottom-Up minimum. · construct an optimal sol. from computed information.
→ It is recursive.	→ It is non recursive
→ It does more work on subproblems and hence has more time consumption.	→ It solves subproblems only once and then stores in the table.
→ It is a top-down approach.	→ Bottom-Up approach.
→ In this subproblems are independent of each other	→ subproblems are interdependent.
→ eg. Merge sort & Binary search ..etc	→ eg. Matrix multiplication