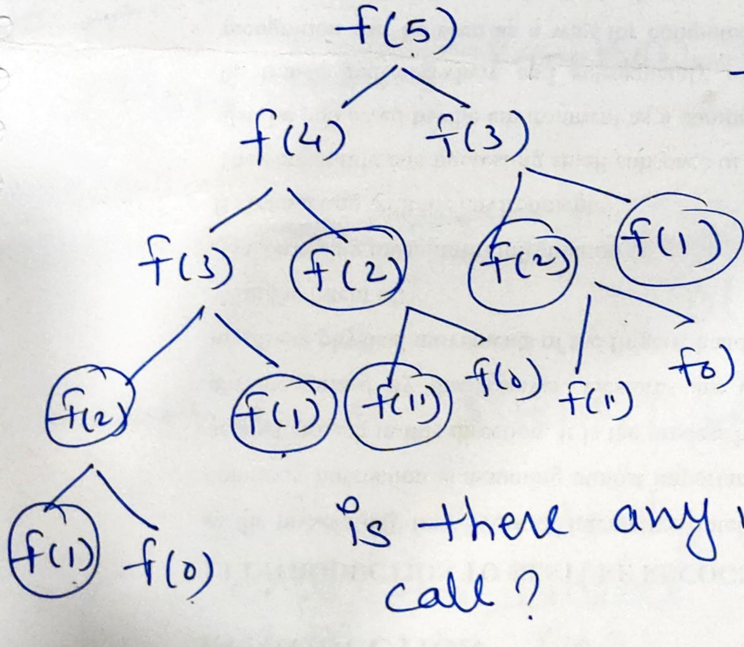
Dynamic programming

$$fib \begin{cases} 0 & \text{if } n=0 \\ 1 & n=1 \\ fib(n-1) + fib(n-2) & n>1 \end{cases}$$

Recursion :-

~~if fib(int n)~~

```
int fib(int n)
{  if (n<=1)
     return n;
   return fib(n-1)+fib(n-2)
}
```



$$T(n) = 2T(n-1)+1$$
$$= O(2^n)$$

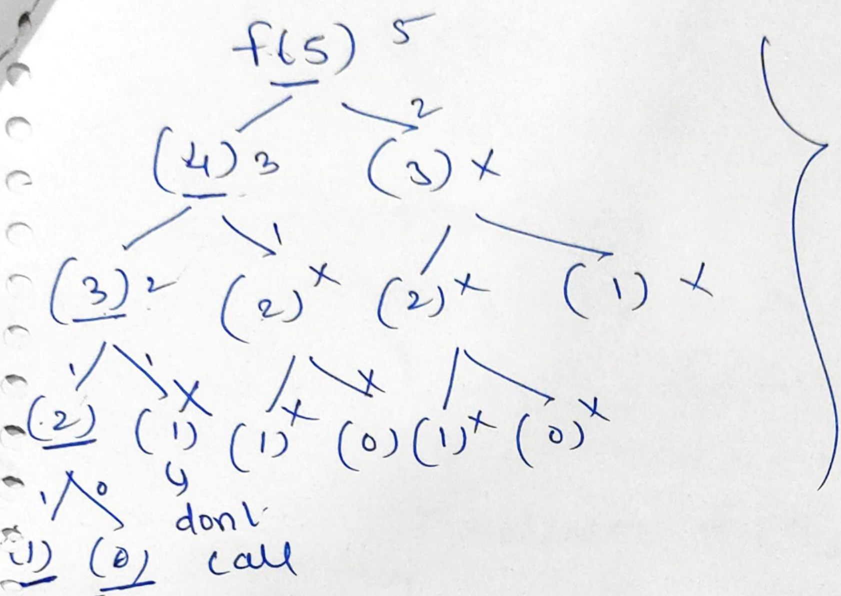
this function is just adding ∴ 1

is there any way to reduce the call?

recursion is also call top down approach

~~-1~~ 0	~~-1~~ 1	~~-1~~ 1	~~-1~~ 2	-1 3	-1 5
0	1	2	3	4	5

Memorization.



Top down approach

How many function calls? only 6

$n+1$

So for fib(n) = n+1 call

$O(n)$

this is Memorization method.

So time complexity is reduced from expo to linear
$O(2^n)$ to $O(n)$

Tabulation method.

iterative method.

```
int fib( int n
{   if(n<=1)
      return n

    f[0]=0, f[1]=1
    for(i=2; i<=n; i++)
    {  f[i]= f(i-1)+f(i-2);

    }  return f(n];
}
```

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

dynamic mostly used iterateive method.

it is bottom up approad starting from 0

Greedy → Predefined procedure to get optimal

dynamic — & we follow the procedure to get the optimal result

dynamic — Try all possible feasible opti solutⁿ & pick up the best solutⁿ

Time consuming

it uses recursive formula

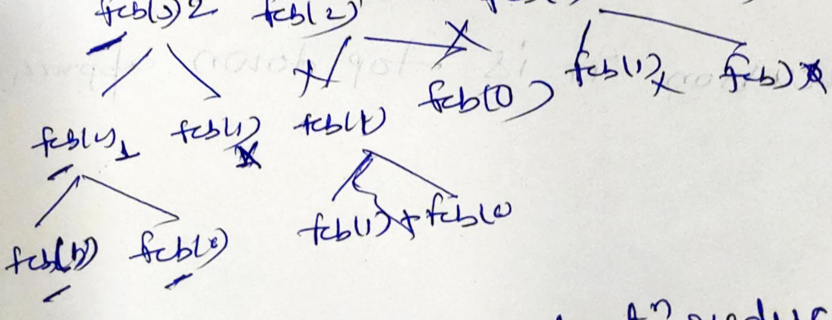
it follow principle of optimality

DY. Prog < Top down — Memorizatⁿ ; bottom up — iterative — dynamic

febno

0	1	1	2	3	5
-1	~~-1~~	-1	-1	-1	-1
0	1	2	3	4	5

$$f(n) = \begin{cases} 0 & \text{if } n=0 \\ 1 & n=1 \\ f(n-1)+f(n-2) & \text{if } i \end{cases}$$



Memorization

Top down approach

only n or n+1 functⁿ call

O(n)

So memorizatⁿ reduces time Complexity. it is a top down approach.

(2) method
iterative approach → tabular approach --

```
int fib(n)
if (n<=1)
  return n
f(0)=0, f(1)=1
for (i=2; i<=n; i++)
{ f(i) = f(i-2) + f(i-1);
}
return f(n);
}.
```

0	1	1	2	3	5

we use iterative approach for ~~tabular~~ method.

~~tabular~~ is a bottom up approach where as recursion it is top down approach

0/1 Knapsack

```
for(i=0; i<=n; i++)
{ for(w=0; w<=m; w++)
  { if (i==0 || w==0)
       k[i][w] = 0
    elseif (wt[i] <= w)
       k[i][w] = max[p[i] + k[i-1][w-wt[i]]
                    , k[i-1][w] ]
    else
       k[i][w] = k[i-1][w];
  }
}
  return k[n][w]
for( i=n ; i>=0 ; i--)
  if (k[i][w] != k[i-1][w])
    { x[i] = 1;
      w = w-wt[i];
    } if w=0
      exit
  else
      x[i]=0;
}
  if w=0
    exit
```

Que

P	1	2	5	6
W	2	3	4	5

m = 8

2 < 8

P	W
1	2
2	3
5	4
6	5

obj

weight

i \ w	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1
2	0	0	1	2	2	3	3	3	3
3	0	0	1	2	5	5	6	7	7
4	0	0	1	2	5	6	6	7	$\boxed{8}$

→ max profit

Arrange it in ↑ order of profit

P	1	2	5	6
W	2	3	4	5

Values inside the table indicate profit

for $i=0, w=0$

$i=1, w=1$

$wt[i] <= w$

$wt[1] <= 1$

$2 <= 1$ No

then

$k[i][w] = k[i-1][w]$

$k[1][1] = k[0][1]$

$= \cancel{1}\, 0$

$i=1, w=3$

$wt[i] = 2$

$wt[i] <= 3$

$2 <= 3$ ✓

$i=1, w=2$, $wt[i]=2$

$wt[i] <= w$

$wt[1] <= 2$

$2 <= 2$ True

$k[i][w] = (p[i] + k[i-1][w-wt[i]], k[i-1][w])$

$= p(1) + k[0][2-2], k[1-1][2])$

$= \max(P[1] + K[0][0],\ K[0][2])$

$= \max(1 + 0,\ 0)$

$= 1$

$i = 2,\ w = 1$

$K[i][w] = K[i-1][w]$

$K[2][1] = K[1][1]$ → copy the value from previous row until $wt[i] \le w$

$i = 2,\ w = 3$

$K[2][3] =$ $wt(i) \le w$ True

$= \max[P[i] + K(i-1)(w - wt(i)),\ K[i-1)(w)]$

$= \max(P[2] + K(2-1)(3-3),\ K(2-1)(3))$

$= \max(2 + K(1)(0),\ K(1)(3))$

$= \max(2 + 0,\ 1)$

$= 2$

$i = 2,\ w = 4,\ wt(i) = 3$

$K[2][4] = \max[P[i] + K[i-1][w - wt[i]],\ K[i-1][w])$

$= \max[P(2) + K[1][4-3],\ K[1][4])$

$= \max(2 + 0,\ 1) = 2$

$i = 2, w = 5, wt[i] = 3$

$$K[2][5] = \max\left[p[i] + K[i-1][w - wt[i]],\ K[i-1][w]\right]$$

$$= \max\left[2 + K[1][2],\ K[1][5]\right)$$

$$= \max[2+1,\ 1)$$

$$= 3$$

$i = 3, w = 5, wt[i] = 4$

$wt[i] \le w$

$$K[3][5] = \max(p[i] + K[i-1][w - wt[i]],\ K[i-1][w])$$

$$= \max(p[3] + K[2][1],\ K[2][5]]$$

$$= \max[5 + 0,\ 3) = 5.$$

max profit = 8

Now which obj is included or not.

for(i = n ; i ≥ 0 ; i--)

if (k[i][w] = = k[i-1][w])

{ cout << i << " = 0" ; i-- ; }

x[i] = 0

else

{ cout << i << " = 1" ; i-- , w - wt[i] ; }

x[i] = 1

if (k(4)(8) = = k(3)(8)) – No

else cout << i << " = 1

4th obj included

Now i-- , w - wt(i)

3 , 8 - 5

3, 3

Goto (3, 3) K(3, 3) = 2

K(3, 3) = = k(2)(3)

& 3rd obj is not included

i-- , i = 2

Goto (2, 3) K(2, 3) = 2

K(2, 3) ≠ K(1)(3)

so 2nd obj is included

i--
i = 1
w - wt(i)
3 - 3
= 0
stop

$M = 165$

$P_1 - P_6 = 100 \quad 50 \quad 20 \quad 10 \quad 7 \quad 3$

$w_1 \quad w_6 - 100 \quad 50 \quad 20 \quad 10 \quad 7 \quad 3$

$M = 6$

$P_1 - P_3 \quad 1 \quad 2 \quad 5$

$w_1 - w_3 \quad 2 \quad 3 \quad 4$

w	i	0	1	2	3	4	5	6
	0	0	0	0	0	0	0	0
2	1	0	0	1	1	1	1	1
3	2	0	0	$\boxed{1}$	2	2	3	3
4	3	0	0	1	2	5	5	$\boxed{6}$

$M = 6$

$6 - 4 = 2$

$x_1 \quad x_2 \quad x_3 \quad x_4$ Arrange profit in order

$1 \quad 0 \quad 0 \quad 1$

$x_1 \quad x_2 \quad x_3$

$1 \quad 0 \quad 1$

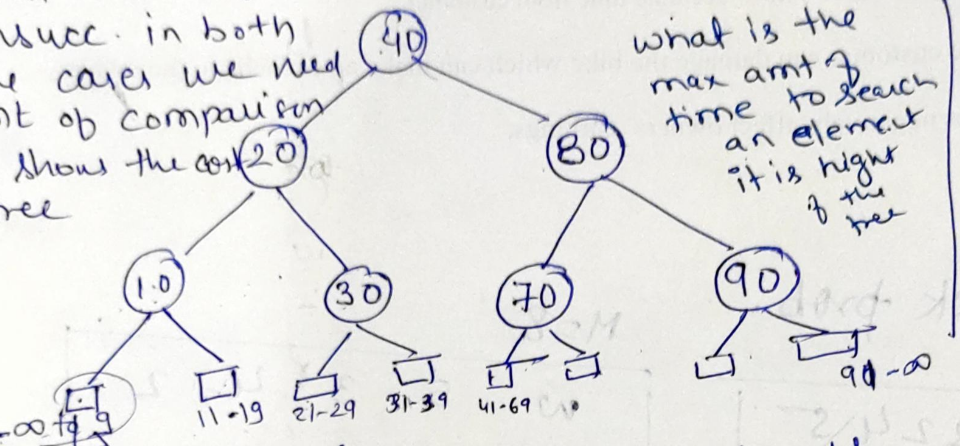
OBST

Optimal Binary Search Tree.

Just start with BST.

for successful & unsucc. in both the cases we need amt of comparison

This shows the cost of tree



what is the max amt of time to search an element it is hight of the tree

level
1
2
3

$\log_2 7 =$

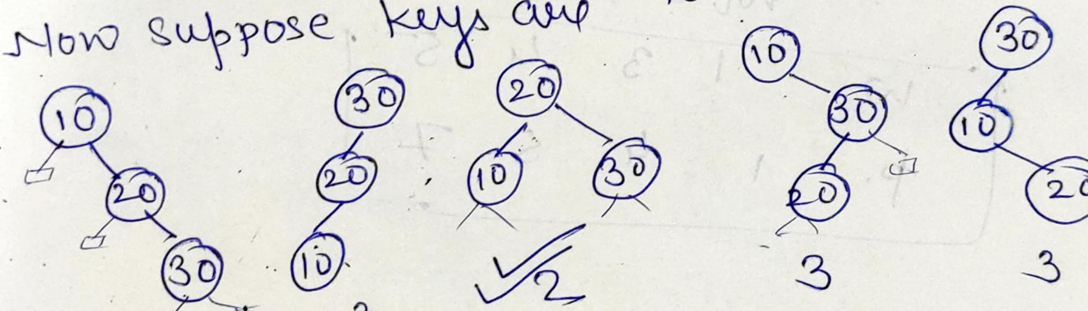
$\log n$

$\frac{1+2+3}{2} = 3$

unsuccessful ~~Max serching~~ req. is

. Max comparison ofor searching = 3

Now suppose. keys are 10, 20, 30



3 comparison 3 ✓2 3 3

we want binary search tree which require less no of comparison

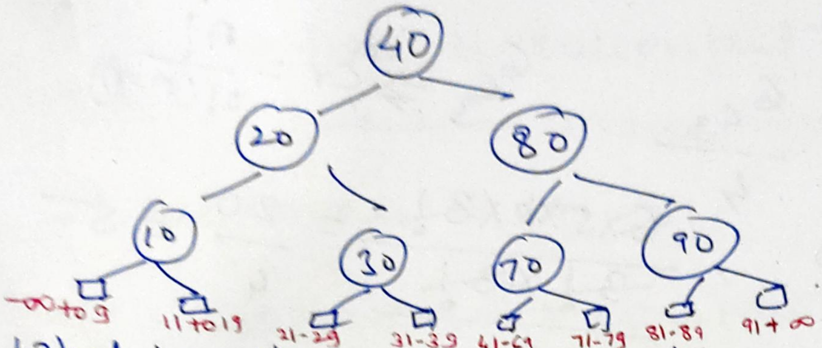
for any key $\underline{\underline{n}}$ there are $\frac{{}^{2n}C_n}{n+1}$ possible trees are available

here $n=3$ so $T(3) = \frac{{}^{2\times 3}C_3}{3+1} = \frac{20}{4} = 5$

$$\left\{ {}^{6}C_3 = \frac{6!}{3! \times (6-3)!} = \frac{6\times5\times4\times3\times2\times1}{3\times2\times1 \times 3\times2\times1} = 20 \right\}$$

OBST

keys. are 40, 20, 10, 30, 70, 80, 90



□ → unsuccessful search
dummy node

logn.

What is the max comparison or time req. to search an element

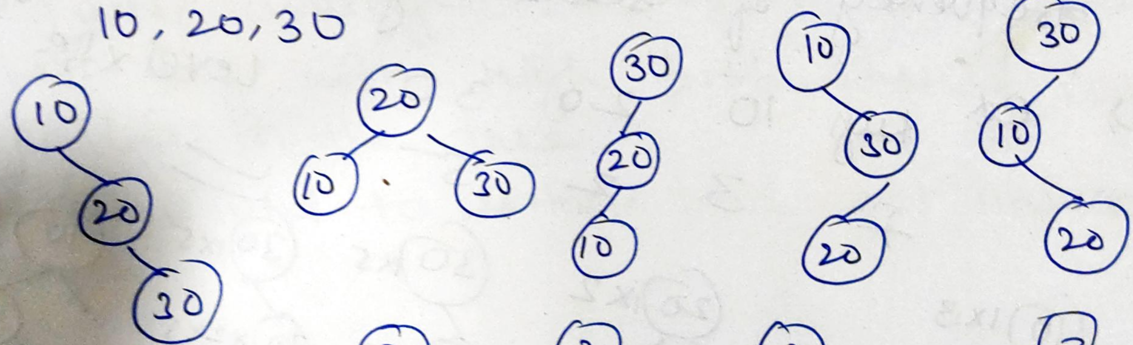
Ans:- it is height of a tree

∴ max 3 comparison

or logn

Now suppose keys are

10, 20, 30



3 comparison   2   3   3   3

we want binary search tree with less comparission

but how many trees we will draw?

for n keys there are $\frac{{}^{2n}C_n}{n+1}$ possible trees are available

$\frac{{}^{2\times3}C_3}{3+1} \Rightarrow \frac{{}^{6}C_3}{4}$ $\quad {}^6C_3 \Rightarrow {}^nC_r = \frac{n!}{r!(n-r)!}$

$\Rightarrow \frac{\frac{\cancel{6\times5\times4\times3!}}{\cancel{3!\times3!}}}{4} = \frac{20}{4} = 5$

possible trees.

but this is not feasible to draw these many trees every time

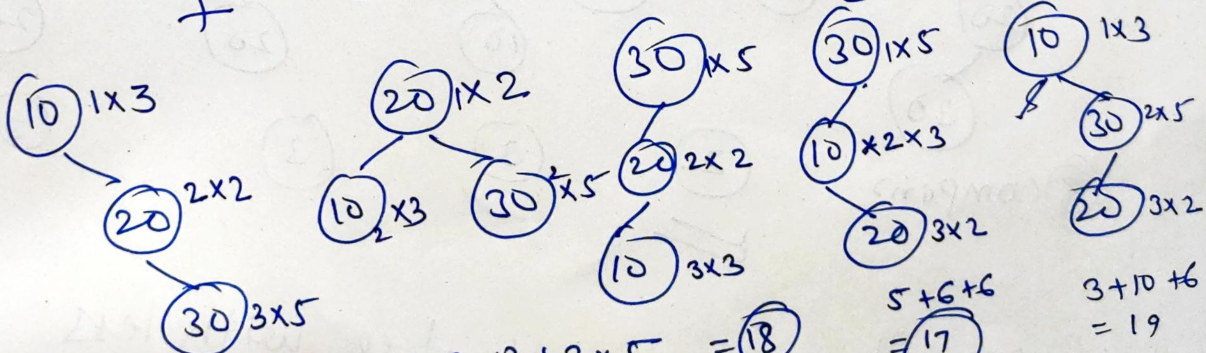
Now what is OBST.

Given a keys and what is the frequency of searching those keys

for ex

key	10	20	30	Level x fre.
f	3	2	5	

So



$3+4+15$
$= 22$

$2+2\times3+2\times5 = 18$
$= 2+6+10$
$= 18$

$5+6+6$
$= 17$

$3+10+6$
$= 19$

This is OBST.
though it is not HBT but

General diff is is search with different prob n there is a prob of unsuccessful search also

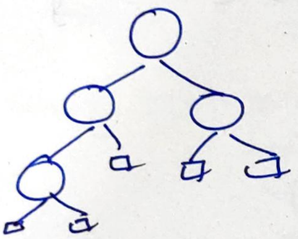
p_i → Prob of successful

q_i → —— unsuccessful

& we know $p_i + q_i = 1$

DP Approach to construct OBST

if we have 4 keys
14 diff. binary seach trees are possible



if there are 4 keys
the 4 successful seach
& (4+1) unsuccessful seach

we will arrange the keys in such a way that key with high prob. should come earlier in the tree.

we also need to ~~se~~ took care abt unsuccessful search.

~~Cost of tree~~

OBST

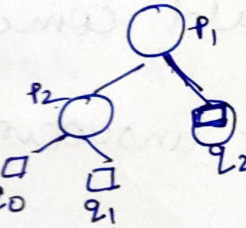
if the keys & their prob. of the keys are Given we have to generate a BST whose cost is min.

cost of tree $\overset{min}{=}$ [cost of left subtree + cost of right subtree] + weight of tree

$$C[i,J] = \min_{i<a\leq J}\{c(i,a-1)+c(a,J)\}+w(i,J)$$

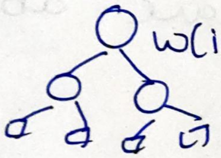
$$w[i,J] = p(J)+q(J)+w(i,J-1)$$

ex. $w[0,2] = \underline{q_0}+\underline{p_1+p_2}+\underline{q_1+q_2}$

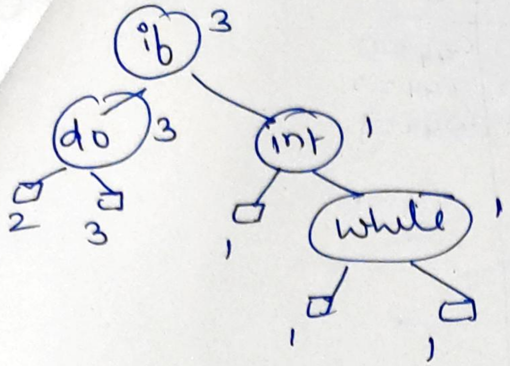


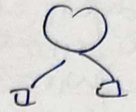
$w[0,3] = \underline{q_0}+\underline{p_1}+\underline{p_2}+p_3+\underline{q_1+q_2}+q_3$

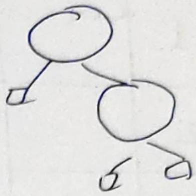
$= w[0,2]+p_3+q_3$



$w(i,J) = w[i,J-1]+p_J+q_J$







$[3\times1 + 3\times2 + 1\times2 + 2\times2 + 3\times2$
$+ 1\times2 + 1\times3 + 1\times3 + 1\times3]$

$= 3+6+2+4+6+2+3+3+3$

$= 10+5+12+5$

$= 10+10+12$

$= 20+12 = 32$

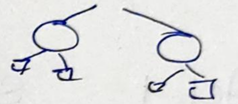
cost-

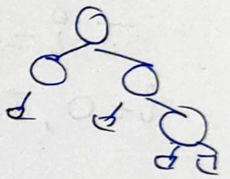
weight- $w(0,1) = 2+3+3 = 8$

$w(0,2) = 8+3+1 = 12$

$w(0,3) = 12+1+1 = 14.$

$w(0,4) = 14+1+1 = 16$



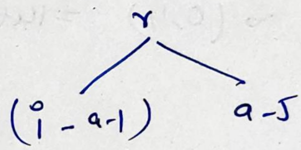


$j-i$ \ j	0	1	2	3	4
0	$c_{00}=0$ $r_{00}=0$ $w_{00}=2$	$c_{11}=0$ $r_{11}=0$ $w_{11}=3$	$c_{22}=0$ $r_{22}=0$ $w_{22}=1$	$c_{33}=0$ $r_{33}=0$ $w_{33}=1$	$c_{44}=0$ $r_{44}=0$ $w_{44}=1$
1	c_{01}	c_{12}	c_{23}	c_{34}	
2					
3					
4.					

Step 1 Set $c(i,J)=0$

$J-i=0$ $\quad c(0,0)=c(1,1)=c(2,2)=c(3,3)=c(4,4)=0$

$i=0, J=0$

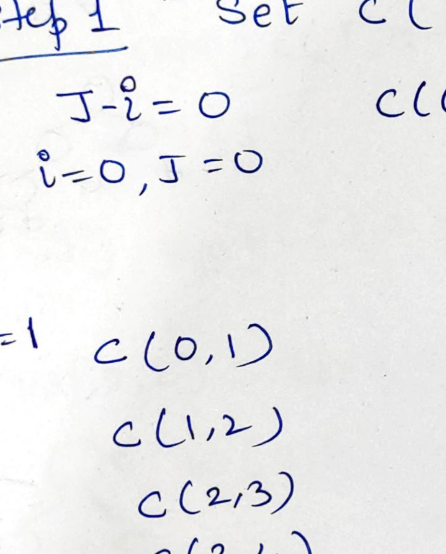


$J-i=1$ $\quad c(0,1)$

$c(1,2)$

$c(2,3)$

$c(3,4)$

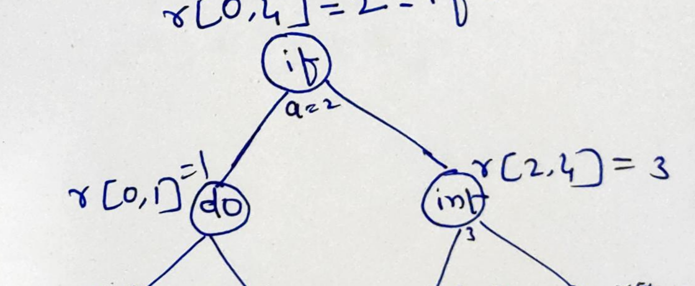


$3\times1+2\times3+2\times1+3\times1$

$=3+6+2+3$

$=\underline{14}$

$r[0,4]=2=$ if.



$r(2,3)=3$ but already we have used 3.

$n = 4$

$$\begin{array}{lll} key = & k_1 - k_4 = & \overset{1}{do}, \overset{2}{if}, \overset{3}{int}, \overset{4}{while} \\ & p(1:4) & (3,3,1,1) \\ & q(0:4) & (2,3,1,1,1) \end{array}$$

Que

Set $c(i,j) = 0$ $\quad j - i = 0$

$r(i,j) = 0$

$w(i,j) = q(i)$

$c(0,0) = c(1,1) = c(2,2) = c(3,3) = c(4,4) = 0$

$r(0,0) = r(1,1) = r(2,2) = r(3,3) = r(4,4) = 0$

$w(0,0) = 2$, $w(1,1) = 3$, $w(2,2) = 1$,

$w(3,3) = 1$, $w(4,4) = 1$

$n = 4$

$i = 0$ ~~to 4~~ $\overset{3}{}$

$j = 1 - 4$

(2) Compute $c(i,j)$ for $j - i = 1$

$w(i,j) = \cancel{w}\ p(j) + q(j) + w(i, j-1)$

root(i,j) = value of a that min $c(i,j)$

$w(i,j)$ for $j - i = 1$ or $\underline{j = i+1}$

$\underline{i = 0}$ $\quad j - 0 = 1 \therefore j = 1$

$w(0,1) = w(0,0) + p(1) + q(1)$

$= 2 + 3 + 3$

$= 8$

$c(0,1) = \min\limits_{0 \le a \le 1} \{ c(i, a-1) + c(a,j) \} + w(i,j)$

$= \min \{ c(0,0) + c(1,1) \} + w(0,1)$

$a = 1$, can not take $a = 0$

$= \min \{ 0 + 0 \} + 8$

$= 8$

root $(0,1) = 1$ {value of a}

$\underline{i=1}$ $J=2$ $J-i=1$

$J=i+1$

$w(1,2) = w(1,1) + P(2) + q(2)$

$= 3+3+1 = 7$

$c(1,2) = \min\{c(i,a-1)+c(a,J)\} + w(i,J)$

$i \le a \le J$ can not take $a=1$ i 1 $a-1$ a $0 \leftarrow 1$

$=$

$\min\{c(1,1)+c(2,2)\} + w(1,2)$

$a=2$

$= \min\{0+0\} + 7 = 7$

$r(1,2) = 2$ {value of a}

$\underline{i=2}$ $J=3$

$w(i,J) = w(i,J-1) + \overline{P(3)} + \overline{q(3)}$

$w(2,3) = w(2,2) + P(3) + q(3)$

$1+1+1 = 3$

$c(2,3) = \min\{c(2,2)+c(3,3)\} + w(2,3)$

$a=3$, can not take $a=2$

$= \min\{0+0\} + 3 = 3$

$r(2,3) = 3$

$\underline{\underline{i=3}}$ $\underline{\underline{J=4}}$ $w(3,4) = w(3,3) + P(4) + q(4)$

$1+1+1 = 3$

$c(3,4) = \min\{c(3,3)+c(4,4)\} + w(3,4)$

$= \underline{3}$

Now For $J-i=2$

$i=0$, $J=2$

$w(0,2) = w(0,1) + P(2) + q(2)$

$8+3+1 = 12$

$c(0,2) = \min\{c(0,0)+c(1,2),\ c(0,1)+c(2,2)\}$

$a = \cancel{0},1,2$ $w(0,2)$

$= \min\{0+7, 8+0\} + 12$

$= 7+12 = 19$

$r(0,2) = 1 \quad (a=1 \text{ min})$

$i=1, J=3$

$w(1,3) = w(1,2) + P(3) + q(3)$

$7+1+1 = 9$

$c(1,3) = \min_{a=1,2,3}\{c(1,1)+c(2,3), c(1,2)+c(3,3)\} + w(1,3)$

$= \min\{0+3, 0+7\} + 9$

$= 3+9 = \underline{12}$

$r(1,3) = 2$

$i=2, \underline{J=4}$

$w(2,4) = w(2,3) + P(4) + q(4)$

$= 3+1+1 = 5$

$c(2,4) = \min_{a=2,3,4}\{c(2,2)+c(3,4), c(2,3)+c(4,4)\} + w(2,4)$

$= \min\{0+3, 3+0\} + 5$

$= 3+5 = 8$

$r(2,4) = \cancel{8}\ 3 \quad (a=3)$

Now for ~~i-J=3~~ $J-i=3$

$\underline{i=0, J=3}$

$w(0,3) = w(0,2) + P(3) + q(3)$

$= 12+1+1 = 14$

$c(0,3) = \min_{a=\cancel{0},1,2,3}\{c(0,0)+c(1,3), c(0,1)+c(2,3), c(0,2)+c(3,3)\} + w(0,3)$

$= \min\{0+12, \underline{8+3}, 19+0\} + 14$

$= 11+14 = 25 \quad \& \ r(0,3)$

$\therefore r(0,3) = \underline{2} \quad (a=2)$

$i=1, J=4$

$w(1,4) = w(1,3) + p(4) + q(4)$

$= 1+1+9 = 11$

$c(1,4) = \min\{ c(1,1)+c(2,4),\ c(1,2)+c(3,4),$
$a = \cancel{1},2,3,4 \quad ,\ c(1,3)+c(4,4)\}$
$+ w(1,4)$

$= \min\{ \underline{0+8}, 7+3, 12+0\} + 11$

$= 8+11 = 19.$

$r(1,4) = 2 \quad (a=2)$

Now for $J-i = 4$

$i=0, J=4$

$w(0,4) = w(0,3) + p(4) + q(4)$

$14+1+1 = 16$

$c(0,4) = \min\{ c(0,0)+c(1,4),\ c(0,1)+c(2,4),$
$a = \cancel{0},1,2,3,4 \quad ,\ c(0,2)+c(3,4),$
$c(0,3)+c(4,4)\}$
$+ w(0,4)$

$= \min\{ 0+19, \underline{8+8}, 19+3, 25+0\} + 16$

$= 16 + 16 = \underline{32}$

for $a=2$

$r(0,4) = \cancel{8}\ \underline{2}.$

2 8

j

iteration ($j-i$)	0	1	2	3	4
0	$w_{00}=2$ $c_{00}=0$ $r_{00}=0$	$w_{11}=3$ $c_{11}=0$ $r_{11}=0$	$w_{22}=1$ $c_{22}=0$ $r_{22}=0$	$w_{33}=1$ $c_{33}=0$ $r_{33}=0$	$w_{44}=1$ $c_{44}=0$ $r_{44}=0$
1	$w_{01}=8$ $c_{01}=8$ $r_{01}=1$	$w_{12}=7$ $c_{12}=7$ $r_{12}=2$	$w_{23}=3$ $c_{23}=3$ $r_{23}=3$	$w_{34}=3$ $c_{34}=3$ $r_{34}=4$	
2	$w_{02}=12$ $c_{02}=19$ $r_{02}=1$	$w_{13}=9$ $c_{13}=12$ $r_{13}=2$	$w_{24}=5$ $c_{24}=8$ $r_{24}=3$		
3	$w_{03}=14$ $c_{03}=25$ $r_{03}=2$	$w_{14}=11$ $c_{14}=19$ $r_{14}=2$			
4	$w_{04}=16$ $c_{04}=32$ $r_{04}=2$				

$j-i = 0$

Left child

Right child

r_{ij}

r → $r_{i,a-1}$, r_{aj}

$r_{04} = 2$ if

r_{02} / $r_{01} = 1$ do $r_{24} = 3$ int

$r_{00} = 0$ $r_{11} = 0$ $r_{22} = 0$ $r_{34} = 4$ while

r_{33} r_{44}

if 3

do 3   int 1

2   3   1   while 1

1   1

r_{04}

r_2

$i \to \underline{a-1}$ aj

$0 \to 1$ 2 4   3

Neha

n 4

Chain Matrix Multiplication

$\frac{{}^6C_3}{4} = \frac{6 \times 5 \times 4}{4 \cdot 2} = 5$

$A_1 A_2 \ldots A_n$ Matrix and we wish to find product diff matrix we need to calculate

$A_1, A_2 \ldots A_n$.

$(((A_1, A_2) A_3) A_4)$ The way of parenthesizing a chain of Matrices

$((A_1 (A_2, A_3) A_4)$ have a dramatic impact on the cost of evaluating the product

$A_1 [2, 3] * A_2 [3, 4]$
$= A_3 [2, 4]$
it require $2 \times 3 \times 4$ no. of multiplication

$A_1 [M \times N] * A_2 [N * P] = A_3 [M, P]$
Require $M * N * P$ no. of Multiplication

Chained Matrix Multiplication prob :-

$A_1 [5 * 3]$, $A_2 [3 * 4]$ $A_3 [4 * 6]$, $A_4 [6 * 5]$

different Parenthesizing gives different no. of multiplication

Q.No.

$(A_1((A_2 A_3) A_4))$ takes -

$(3 * 4 * 6) + (3 * 6 * 5) + 5 * 3 * 5$

$= 237$ Multiplicat?

$(A_1 (A_2 (A_3 A_4)))$ takes 255 Multi

$(A_1 A_2)(A_3 A_4))$ takes 280 M.

$(((A_1 A_2) A_3) A_4)$ takes 330 M.

$((A_1 (A_2 A_3) A_4)$ →

$A_1 \; A_2 \; A_3 \; A_4$ takes 312

$A_3 \; A_4$

n no. of matrices - A

Solution - divide the prob into subprob
Here we use principle of optimality which is said to apply if an optimal solu" to an instance of a prob always contain optimal solu" to all subinstance

if $(A_1 (A_2 A_3) A_4)$ is optimal order then we know
$(A_2 A_3) A_4$ is optimal order for $A_2 A_3 A_4$

→ We need to find best way to paranthsize the chain of matrics to minimize the no of scalar multi.
This can be done by dividing the prob into subprob and finding

Q.No.					

optimal soluⁿ to the subprob.

eg:- A_1 — A_n chain Matrix

The chain is divided into subprob (A_1 — A_k) and (A_{k+1} — A_n)

but the prob is find the value of k.

we can find k by looking at the optimal soluⁿ to each of subprob

N[1][1] hold the no. of multiplicatⁿ to multiply from A_1 to A_1. This is chain of length 1. N[i][i]=0 i=1 to n

N[1][2] — " — A_1 to A_2 this is chain of length 2

A1.A2

Add N[2][3], N[3][4] are also chain of length 2

N[1][3] — " — A_1 to A_3. it is chain of length 3

it can be either $(A_1 A_2)A_3$ or $A_1(A_2 A_3)$

hence we have 2 possibilities. so we choose minimum. we use the term $\underline{K}$ to show where we decide to put parantheses

$$Q_{i,j} = \min\{Q_{ik} + Q_{k+1,j} + d_{i-1} * d_k * d_j\}$$

where $i \leq k < j$

$$C_{ij} = \min\{C_{ik} + C_{k+1,j} + d_{i-1} * d_k * d_j\}$$

P values can be used to find parenthesis order of $\boxed{\text{matrices} \mid \text{recursively} \mid A_1 \text{ to } A_n}$

$A_{1..n} = (A_1 \ldots P[1..n], A[P[1,n]+1] \ldots n)$

Ex:-

Matrix	A_1	A_2	A_3	A_4	A_5	A_6
dimension	30x35	35x15	15x5	5x10	10x20	20x25
	30.35	(35x5)				

Step 1. Convert 2D into 1D

1D Array

Index	0	1	2	3	4	5	6
d	30	35	15	5	10	20	25

2D Array N and P upper value denotes N[i][j] and lower value denotes P[i][j]

P[i][j] holds value of K which minimizes N[i][j]

Index	J=1	J=2	J=3	J=4	J=5	J=6
i=1	0	15750 K=1	7875 K=1	~~4375~~ 9375 K=3	11875 K=3	15125 K=3
i=2		0	2625 K=2	4375 K=3	7125 K=3	10500 K=3
i=3			0	750 K=3	2500 K=3	5375 K=3
i=4				0	1000 K=4	3500 K=5
i=5					0	5000 K=5
i=6						0

Chain Matrix continue

Initialization — ~~for i = 1 to n & for J = 1 to n~~ do

No. of Matrix $n = 6$ $N[i\,i] = 0$

$J - i = 0$

$N[1][1] = 0$ $N[3][3] = 0$

$N[2][2] = 0$ $N[4,4] = 0$

$N[5][5] = 0$ $N[6,6] = 0$

step 1 for $i = 1$, $J = i+1$ compute $N[i][J]$

do this until $\underline{J - i = N-1}$

Step 1 — Compute values. for $i \le K < J$

for $i = 1$ $K = 1$, $J = 2$ $\{ J - i = 1$

$$N[1][2] = \min[\underset{i}{N[1]}\underset{K}{[1]} + N\underset{K+1}{[2]}\underset{J}{[2]} + d_0 * d_1 * d_2]$$

$$= 0 + 0 + 30 * 35 * \cancel{35}\,15$$

$$= 15750$$

$K = 1$

for $i = 2$, $K = 2$, $J = 3$

$$N[2][3] = \min[N[2][2] + N[3][3] + d_1 * d_2 * d_3$$

$K=2$

$$= 0 + 0 + 35 * 15 * 5$$

$$= 2625$$ $K=3$

Q.No.

for $i=3$, $K=3$, $J=4$

$N[3][4] = \min[N[3][3] + N[4][4] + d_2 * d$
$= 0+0+15*5*10$
$= 750$

$i=4$, $K=4$, $J=5$

$N[4][5] = \min[N[4][4] + N[5][5] + d_3 * d$
$= 0+0+\cancel{10} \times 5 \times 10 \times 20$
$= 1000$

$i=5$, $K=5$, $J=6$

$N[5][6] = \min[N[5][5] + N[6][6] + d_4 * d$
$= 0+0+10 \times 20 \times 25$
$= 5000$

Step 2 Compute values for $i \le K < J$

for $i=1$, $J=3$, $1 \le K < 3$ $\{J-i=2$
$K=1,2$

$N[1][3] = \min[N[1][1] + N[2][3] + d_0 * d_1 * d_3,$
$N[1][2] + N[3][3] + d_0 * d_2 *$

$= \min\{[0+2625+30*35*5,$
$15750+0+30*15*5]\}$

$= \min\{7875, 18000\}$
$= 7875$ for $K=1$

Q.No.

for $i=2, J=4, K=2,3$

$N[2][4] = \min\{N[2][2] + N[3][4] + d_1 * d_2 * d_4, N[2][3] + N[4][4] + d_1 * d_3 * d_4\}$

$= \min\{6000, 6375\}$

$K=3$

$N[3][5] = \cancel{\min\{0+1000+15\times5\times20\}}$

$= \min(2500, 3750)$

$= 2500$ for $K=3$.

$N[4][6] = \min(6256, 3500)$

$= 3500 \quad K=5$

Step 3 Compute values for $i \le K < J$

$K = 1,2,3$

$N[1][4] = \min\{14875, 21000, 9375\}$

$J-i=3$

$= 9375 \quad K=3$

$N[2][5] = \min\{13000, 7125, 11375\}$

$K=2,3,4$

$= 7125 \quad K=3$

$N[3][6] = \min\{5375, 9500, 10006\}$

$K=3,4,5$

$= 5375 = K=3$

Q.No.

Step4 $i=1, J=5, k=1,2,3,4$
$J-i=4$
$N[1][5] = \min\{28125, 27250, 11875, 15375\}$
$= 11875$ for $K=3$

$i=2 \quad J=6 \quad K=2,3,4,5$

$N[2][6] = \min\{18500, 10500, 18125, 24625\}$
$= 10500$ for $K=3$

Step5 $i=1, J=6, K=1,2,3,4,5$

$N[1][6] = \min[36750, 32375, 15125, 21875, 26875]$
$J-i=5$
$= 15125$ for $K=3$

Matrix Multiplicatⁿ chain can be obtained by using $A_{1..n} = (A_{1...P[1..n]}, A_{P[1,n]+1..n})$ as follow

$A_{1..6} = (A_{1..P[1,6]}, A_{P[1,6]+1...6})$
$= (A_{1...3} \; A_{4...6})$ as $P[1,6]=3$
$= ((A_{1...P[1,3]} \; A_{P[1,3]+1...3})(A_{4..P(4,6]} \; A_{P[4,6]+1..6}))$

$(A_1\ A_2\ A_3)(A_4\ A_5\ A_6)$

$(A_1)(A_2.A_3)\ (A_4\ A_5)(A_6)$

MIT | Academy of Engineering

Alandi, Pune - 412 105.

CLASS TEST / MID - SEMESTER / PRELIMINARY EXAMINATION

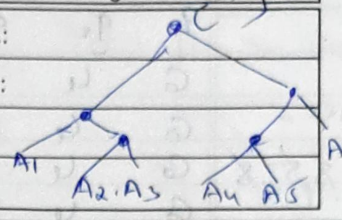
TERM I / II - 20 - 20

22

Supervisor's Signature & Date

A[1-6]

Name : $((A_1 - A_3).(A_4 - A_6))$	Roll No.:
Year : $((A_1.(A_2.A_3)).((A_4.A_5)).A_6)$	Branch :
Subject :	Date :
Section :	Marks :



$\Rightarrow ((A_{1..1}\ A_{2..3})\ (A_{4..5}\ A_{6..6})$

back back

as $P[1,3] = 1 \quad P[4,6] = 5$

$\Rightarrow ((A_1\ A_{P(2..3)}\ A_{P[2..3]+1}\ ...3))$

$((A_4\ ...\ P[4,5]\ A_{P[4-5]+1}\ ...5)\ A_6))$

$P[2,3] = 2$

$= ((A_1\ (A_{2..2}\ A_{3..3}))\ ((A_{4..4}\ A_{5..5})\ A_6))$

$((A_1\ (A_2\ A_3))\ ((A_4\ A_5)\ A_6))$

is optimal solu^n

$(A_1 - A_6)$

$(A_1\ A_2\ A_3\ A_4\ A_5\ A_6)$

$((A_1 A_2 A_3)(A_4 A_5 A_6))$

$(((A_1)(A_2 A_3))\ ((A_4 A_5) A_6))$

$(A_1 - A_3)\ ($

A[1,6]

A[1-3] A[4-6]

[A(1) A[2,3]] [A[4-5] A[6]]

[A[1]. A[2][3]] [A[4] A[5] A[6]]

$((A_1 . A(2,3)) . (A(A4.A5) . A(6))$

~~30.2625~~

A[1-6]

A[1-3] . A[4-6]

$A_1\ [A_2 - A_3) . A(4-5) . A6$

$((A_1 (A_2 . A_3)) . ((A_4 . A_5) A_6))$

	A_1	A_2	A_3	A_4		
	3,2	2,4	4,2	2,5		
Index	0	1	2	3	4	5
d	3	2	4	2	~~5~~	5

$N[2,4] = \min[N[2,2]+$
$k=2,3$ $N[3,4]+d_1*d_2*d_4,$

~~$= \min[0+40+2\times4\times5,$~~

$N[2,3]+N[4,4]+d_1*d_3*d_4]$

$= [0+40*2*4\times5,$
$16+0+2\times3\times5]$

$= [40+40, 16+30]$

$= \min(80, 46)$

$=$ ~~46~~ 36 at $k=3$

$i=1, j=4$ $j-i=3$

$N[1,4] = \min[N[1,1]+$
$k=1,2,3$ $N[2,4]+d_0*d_1*$
$N[1,2]+N[3,4]+d_0*d_2*$
$N[1,3]+N[4,4]+d_0*d_3*$

$= \min[0+36+3*2*5,$
$24+40+3*4*5,$
$28+0+3*2*5]$

$= \min [36+30, 64+60, 28+30]$

Q.No. $= \min(66$	124,	$58)$		

$n=4$

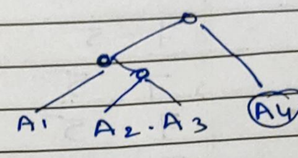
$N[1,1], N[2,2], N[3,3], N[4,4] = 0$

	j=1	j=2	j=3	j=4
i=1	0	24, k=1	28, k=1	58, k=3
i=2		0	16, k=2	36, k=3
i=3			0	40, k=3
i=4				0

$(A_1 A_2 A_3 A_4)$

$((A_1 A_2 A_3) A_4)$

$((A_1)(A_2 . A_3))(A_4))$



$N[1,2] \quad i = J = 2 \quad J = i+1$

$k=1$

$N[1,2] = \min(N[1,1] + N[2,2] + d_0 * d_1 * d_2)$

$= [0+0+3*2*4]$

$= 24 \quad k=1$

$d_{i-1} * d_k * d_j$

$N[2,3] = \min[N[2,2] + N[3,3] + d_1 * d_2 * d_3]$

$k=2$

$= \min[0+0+2*4*2]$

$= 16$

$N[3,4] = \min[N[3,3] + N[4,4] + d_2 * d_3 * d_4]$

$k=3$

$= \min[0+0+4 \times 2 \times 5]$

$= 40$

2nd iteration

$i=1, J=3 \quad J-i=2$

or $J=i+2$

$N[1,3] = \min [N[1,1] + N[2,3] + d_0 * d_1 * d_3,$
$k=1,2 \quad N[1,2] + N[3,3] + d_0 * d_2 * d_3]$

$= \min[0+16+3 \times 2 \times 2,$
$24+0+3 \times 4 \times 2]$

$= \min[28, 48]$

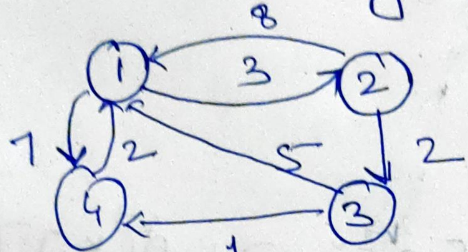
$= 28$ at $k=1$

All pair shortest path

Floyd warshall Algo

it can not handle (-)ve weight cycle

→ Prob:- find the shortest path from b/w every pair of vertices



$$A^0 = \begin{array}{c|cccc} i \backslash j & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & \infty & 7 \\ 2 & 8 & 0 & 2 & \infty \\ 3 & 5 & \infty & 0 & 1 \\ 4 & 2 & \infty & \infty & 0 \end{array}$$

∞ → No direct path.

n = 1, 2, 3, 4

Assume vertex 1 is the intermediate vertex.

$$A^1 = \begin{array}{c|cccc} i \backslash j & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & \infty & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 12 \\ 4 & 2 & 5 & \infty & 0 \end{array}$$

find the value of $A_{23}, A_{24}, A_{32}, A_{34}, A_{42}, A_{43}$ via vertex 1

if $A^0[2,3] < A^0[2,1] + A^0[1,3]$

$2 < 8 + \infty$

$2 < \infty$

then $A^1[2,3] = A^0[2,3]$

$A^0[2,4] < A^0[2,1] + A^0[R,4]$

$\infty < 8 + 7$

$\infty \nless 15$

then $A^1[2,4] = A^0[2,1] + A^0[1,4]$

15

Formula

$$A^k[i,j] = \min[A^{k-1}[i,k] + A^{k-1}[k,j],\ A^{k-1}[i,j]]$$

```
for ( k=1 ; k<=n; k++ ) => intermidiate
{
  for( i=1; i<=n; i++ )
  {
    for ( J=1 ; J<=n; J ++ )
    {
      A^k[i,J] = min (A^(k-1)[i,k] + A^(k-1)[k,J],
                      A^(k-1)[i,J] ]
    }
  }
}
```

$O(n^3)$

$i \rightarrow k \rightarrow J$

$1 \rightarrow 1 \rightarrow 1$

$1 \rightarrow 1 \rightarrow 2$

$$A^2 = \begin{bmatrix} 0 & 3 & 5 & 7 \\ 8 & 0 & 2 & 15 \\ 5 & 8 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{bmatrix}$$

k=2 as intermediate vertex
base matrix will be A' matrix
keep 2nd row & 2nd column of A' matrix as it is & diagonal

Find value of $A13$, $A14$
$A31$, $A34$
$A41$ $A43$

$A'[1,3] \leq A[1,2] + A[2,3]$
$\infty = 3 + 2$
$= 5$

$A^2[1,4] = \min[A'[1,4], A'[1,2] + A'[2,4]]$ vertex
$= \min(7, 3+15)$
$= \underline{\underline{7}}$

K=3

$$A^3 = \begin{bmatrix} 0 & 3 & 5 & 6 \\ 7 & 0 & 2 & 3 \\ 5 & 8 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 0 & 3 & 5 & 6 \\ 5 & 0 & 2 & 3 \\ 3 & 6 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{bmatrix}$$

Single source shortest path.

Bellman ford → handle (-)ve weight

Similar to Dijkstra Algo
↳ this can not handle
(-)ve weight i/p

b

can not handle (-)ve weight cycle
here u need to do relexation of edges

$u \xrightarrow{c(u,v)} v$

~~if (d[u] + d[v] >~~

~~d[u,v]~~ if (d[u] + c(u,v) < d[v])
then d[v] = d[u] + c(u,v)

steps :-

(1) Initialization :- Set distance of source vertex to all the remaining vertex to ∞
& dis[source] = 0

(2) cal shortest path by Relexation of edges |V-1| times
d[u] + c(u,v) < d[v]
then d[v] = d[u] + c(u,v)

(3) hande (-)ve weight cycle
if d[v] > d[u] + c(u,v) then it contain (-)ve weight cycle

Iteration	Vertex 1	2	3	4	5
0	0	∞	∞	∞	∞
1	0	~~6~~ 3	5	5	8
2	0	3	5	2	~~8~~ 5
3	0	3	5	2	5
4					
5					

Stop

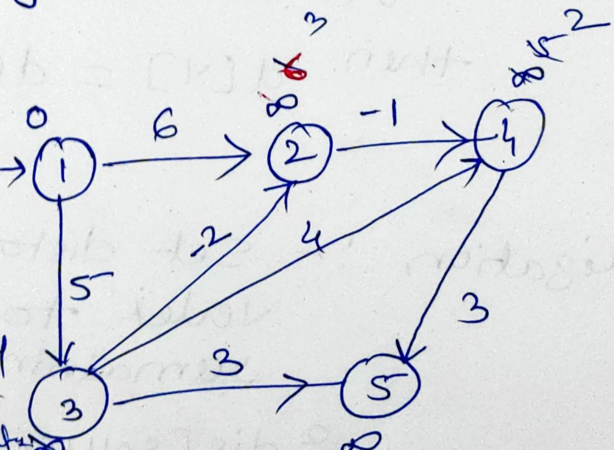
Time complexity → e, edges has to be relaxed $|V-1|$ times

So $|e| \times |V-1|$

$|e| \times |V|$

$n \times n$

$= O(n^2)$



if the Graph is complete i.e each vertex is directly connected to all other vertex then the complexity is edges · vertex

$\frac{n \cdot (n-1)}{2} \cdot (n-1)$

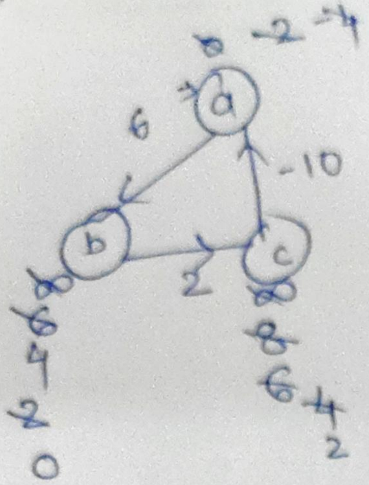
$= O(n^3)$

edges (1,2) (2,4) (3,2) (4,5)
(1,3) (3,4)
(3,5)

7 edges

relax $|V-1|$ time

means 4 time



3 edges

ab, bc, ca

3 edges so do relax 2 time

4th time also edges were relaxed. So the graph is having a (-)ve weight cycle. this can not be handle by BF algo.