For B_i at k^{th} level we have ${}^{i}C_{k}$ number of nodes.

→ Binomial heaps is a collection of binary tree.

* ⇒ Disjoint Set Data Structure.

3 operations performed by ~~Dat~~ Disjoint Set DS

i) Make-Set
ii) Find-Set
iii) Union.

i) Make set - It will convert set elements into singleton set (For e.g. {1,2,3} → {1},{2},{3}
For function call write.
Make-Set(1) or Make-Set(2) or Make-Set(3))

ii) Find-Set :- It will return the representative element of the particular ~~set~~ element

iii) Union :- It will be applicable only when Find-Set of 2 elements are different.

For e.g. X = {a,b,c}
Y = {x,y,z}

For X, let {a} is the representative
For Y, let {x} " " "

~~As X~~ For same set elements we cannot do union as any ~~the~~ particular set has only 1 representative.

Not possible → Union of (Find-Set(a), Find-Set(b))
Possible → Union of (Find-Set(a), Find-Set(x))

Date:

#

⇒ Applications of Disjoint - Set DS

I) Finding connected components in a graph (undirected)

→ Connected components means between 2 vertices at least 1 path should be there.

→ For Algorithm we will ~~do~~ go edge by edge in line number 3, 4, 5

→ ~~e~~ e & g are different so they can be unioned

→ e, g are now joined consider any one as representative ~~& they are joined with f.~~

II) See ppt

⇒ List of Paradigms

1) Backtracking
2) Branch & bound
3) Brute-force search
4) Divide & conquer
5) Dynamic programming
6) Greedy Algorithm
7) Prune & search.

Non - Conflicting Activities

→ 2 activities are said to be non-conflicting if $s_i \geq f_j$ or $s_j \geq f_i$ where s_i & s_j denote the starting time of activities i & j respectively & f_i & f_j refer to the finishing time of the activities i & j respectively.

→ Time complexity will be N-logn

Eg.

	0	1	2	3
S	0	3	5	11
F	4	7	8	13

Activity 0, 2, 3 will be selected
For activity 1 $F_0 > S_1$ ∴ Activity 1 will not be selected
For activity 2, $F_0 < S_2$ ∴ 2 will be selected
$F_2 < S_3$ ∴ 3 will be selected.

⇒ Job sequencing with deadline

We will arrange all the profits in non-increasing order.

→ Refer gatevidyalay website.

Eg.

Jobs	J_1	J_2	J_3	J_4	J_5	J_6
Deadlines	5	3	3	2	4	2
Profits	200	180	190	300	120	100

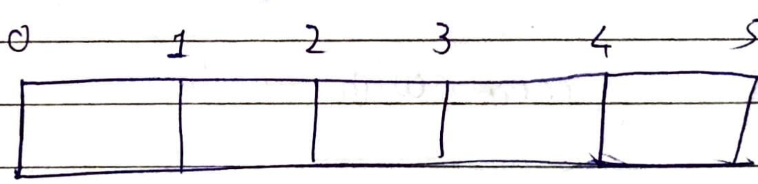
Sol^n

Descending order of profit

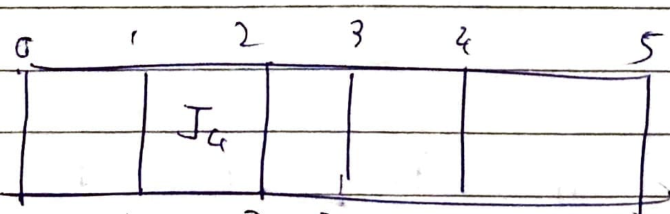
J4, J1, J3, J2, J5, J6

Jobs	Dead line	Profit
J_4	2	300
J_1	5	200
J_3	3	190
J_2	3	180
J_5	4	120
J_6	2	100

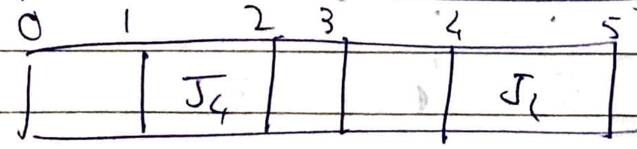
→ Now J_1 has got more time i.e. 5 days or 5 week comparatively to other jobs.



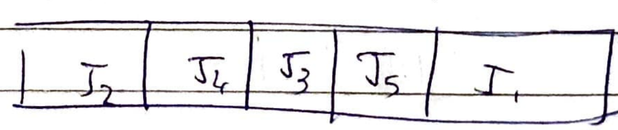
→ You have to keep your job initially as far as possible. in deadline



Now



Now



Page No.
Date:

⇒ Matrix chain Multiplication

$\quad P_0 \; P_1 \; P_2 \; P_3 \; P_4 \; P_5$

Ex:- < 4, 10, 3, 12, 20, 7 >

$n = 5$

$\hspace{6em} 0$

$m[1,5] = m[1,1] + m[2,5] + P_0 P_1 P_5$ — $k=1$ $= 1350+280 = 1680$

$= m[1,2] + m[3,5] + P_0 P_2 P_5$ — $k = 2$ = 120 + 1140 + 84 =

$= m[1,3] + m[4,5] + P_0 P_3 P_5$ — $k = 3 = 264 + 1680$

$= m[1,4] + m[5,5] + P_0 P_4 P_5$ — $k = 4 = 1080 + 560$

$\hspace{6em} 0 \hspace{12em} =$

$k=2$

$m[2,5] = m[2,2] + m[3,5] + P_1 P_2 P_5$ $\quad k = 2 = 1140 + 210 = 1350$

$m[2,3] + m[4,5] + P_1 P_3 P_5$ $\quad k = 3 = 360 + 1680 +$

$m[2,4] + m[5,5] + P_1 P_4 P_5$ $\quad k = 4 = 1320 + - + 1400$

$m[3,5] = m[3,3] + m[4,5] + P_2 P_3 P_5 = 1680 + 252 = 1932$

$= m[3,4] + m[5,5] + P_2 P_4 P_5$ $k=4$ $= 720 + 420 = 1140$

$m[4,5] = P_3 P_4 P_5 = 1680$

$m[3,4] = P_2 P_3 P_4 = 720$

$m[2,3] = P_1 P_2 P_3 = 360$

$m[1,2] = P_0 P_1 P_2 = 120$

save the value

$m[1,3] = m[1,1] + m[2,3] + P_0 P_1 P_3$ — $480 + 360 = 840$

$= m[1,2] + m[3,3] + P_0 P_2 P_3$ $k=2$ — $120 + 144 = 264$

$m[1,4] = m[1,1] + m[2,4] + P_0 P_1 P_4 = 1320 + 800 = 2120$

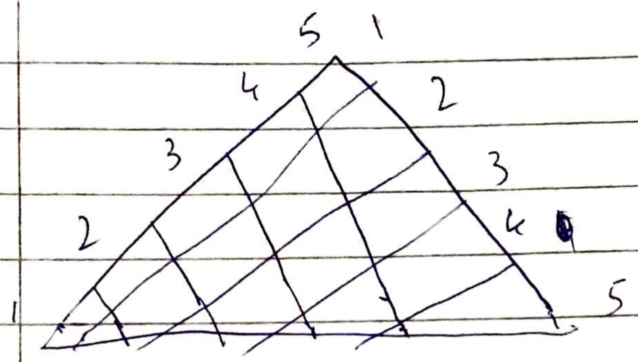
$= m[1,2] + m[3,4] + P_0 P_2 P_4$ $k=$ $120 + 720 + 240 = 1080$

$= m[1,3] + m[4,4] + P_0 P_3 P_4 = 264 + 960 = 1224$

$\boxed{k=2}$

$m[2,4] = m[2,2] + m[3,4] + P_1 P_2 P_4 = 720 + 600 = 1320$

$= m[2,3] + m[4,4] + P_1 P_3 P_4 = 360 + 2400 = 2760$



$k=3$ $\quad$ $K=1$

$s[1,6] \rightarrow s[1,3] \rightarrow s[1,1]$

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

m[1,1]	m[2,2]	m[3,3]	m[4,4]	m[5,5]
m[1,2]	m[2,3]	m[3,4]	m[4,5]	
m[1,3]	m[2,4]	m[4,5]		
m[1,4]	m[2,5]			
m[1,5]				

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

$\boxed{\text{For right side}}$ $k=5$

Now $s[4,6]$

$\rightarrow$

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

$k=5$

Now $s[5,6]$

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

So for left side we have to decrement J

& for right side we have to increment I

⇒ Longest Common Subsequence

A subseq of a given sequence is just a sequence with one or more char left out.

Eg:- X :- BDCABA
Y :- ABCBDAB

As here string is matched we have put 1.

		B	D	C	A	B	A
	0	0	0	0	0	0	0
A	0	0	0	0	1	1	1
B	0	1	1	1	1	2	2
C	0	1	1	2	2	2	2
B	0	1	1	2	2	3	3
D	0	1	2	2	2	3	3
A	0	1	2	2	3	3	4
B	0	1	2	2	3	4	4

Initially add extra rows of 0.

BCBA

~~A~~

BCAB

~~BCBA~~

BA

~~ABBA~~

For match

a	
	a+1

~~For no match~~

	c
b	

→ b or c (bigger one)

For moving either left or top. and write answer from right to left

Ex

<1, 0, 0, 1, 0, 1, 0, 1>

<0, 1, 0, 1, 1, 0, 1, 1, 0>

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