18/4/2/6

Dhiraj Bodake

Tutorial No-4

Q.1) There exists a lecture hall in which the lecture have to be arranged. The main condition is such that now two lectures should be overlapped. I/P consists of starting time & Finishing time of every lecture such that maximum lectures takes place in the hall

I/P - Data

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}
St	0	3	1	3	6	5	5	2	12	8	8
Ft	6	5	4	8	10	7	9	13	14	11	12

Greed - max lecture

constraint - no two lectures should be overlapped.

→ Approach :-

- sort all pairs (lectures) in increasing order of second number (Finishing time) of each pair

- Select First lecture of sorted pair as First lecture in the hall and push it into result vector and set a variable time limit with the second value (Finishing time) of First selected lecture.

- The second pair to last pair of array and if the value of First element (starting time of lecture) of current pair of greater.

select the current pair and update the result vector (push selected lecture number into vector) and variable time_limit.

- on printing the order of lecture from vector we get lectures that are conducted without overlap.
- size of vector is the no. of maximum lecture possible.

eg –
after sorting the given input pair in increasing order of finishing time.

	L3	L2	L1	L6	L4	L7	L5	L10	L11	L8	L9
st	1	3	0	5	3	5	6	8	8	2	12
ft	4	5	6	7	8	9	10	11	12	13	14

- first lecture that is possible is L3 which finished ~~on~~ after 4.
- second lecture that is possible is L6 as it start after L3 finish.
- third lecture that is possible is L10 as it start after L6 finish.
- fourth lecture that is possible is L9 as it start after L10 finish.
- After lecture L9 no lecture is possible so, maximum no. of lecture possible are 4.
  i.e. (L3, L6, L10, L9)

GoodLuck Page No.
Date

Q.2) capacity of knapsack $M=15$, no. of objects $n=7$.

objects	1	2	3	4	5	6	7
profit	10	5	15	7	6	18	3
weight	2	3	5	7	1	4	1

→ Here we put object into knapsack according to ~~increasing order~~ of decreasing order of profit/weight ratio.

$\frac{P}{W}$ = 5 1.3 3 1 6 4.5 3

x = 1 $\frac{2}{3}$ 1 0 1 1 1

steps :-

1) object 5 with fraction $x=1$,
rem_weight $= 15-1 = 14$
2) object 1 with fraction $x=1$,
rem_weight $= 14-2 = 12$
3) object 6 with fraction $x=1$,
rem_weight $= 12-4 = 8$
4) object 7 with fraction $x=1$
rem_weight $= 8-1 = 7$
5) object 3 with fraction $x=1$
rem_weight $= 7-5 = 2$
6) object 2 with fraction $x=\frac{2}{3}$
rem_weight $= 2-2 = 0$
7) all remaining object contribute $x=0$ fraction.

Total profit $= (10\times1) + (5\times\frac{2}{3}) + (15\times1) + (6\times1) + (18\times1) + (3\times1)$
$= 10 + 3.3 + 15 + 6 + 18 + 3$
$= \cancel{56} \; 55.3$

Q.3)

jobs	j1	j2	j3	j4	j5	j6	j7
profit	3	5	20	18	1	6	30
Deadline	1	3	4	3	2	1	2

→ prefer the job with most profit.

slot hours → 0 — 1 — 2 — 3 — 4

Jobs → ↑ ↑ ↑ ↑

j6 (6) j7 (30) j4 (18) j3 (20)

profit = 6 + 30 + 18 + 20 = 74

approach:-

- sort all jobs in decreasing order of profit.
- Iterate on jobs in decreasing order of profit. For each job, ~~do~~ find an empty time slot from deadline to 0. If found empty slot put the job in the slot and mark this slot filled.

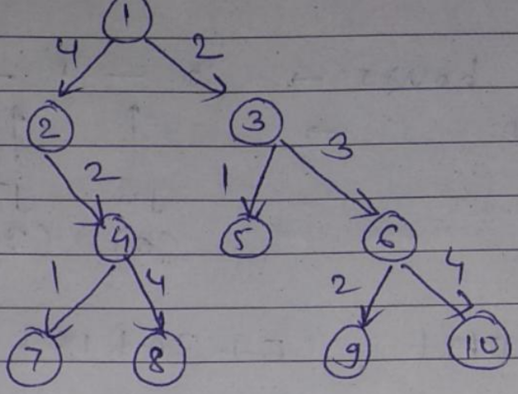
example:-

after sorting in decreasing order of profit.

jobs	j7	j3	j4	j6	j2	j1	j5
profit	30	20	18	6	5	3	1
Deadline	2	4	3	1	3	1	2

for ~~slot~~ job j7, slot 1-2 get filled.
for job j3, slot 3-4 get filled.
for job j4, slot 2-3 get filled
for job j6, slot 0-1 get filled.
as above and we get profit =74.

Q.4) for the tree of figure solve the TVSP when ① $\delta = 4$ ② $\delta = 6$



→ ~~① for $\boxed{\delta = 4}$~~

Here we use greedy approach that is if u has a parent v such that $d(u) + w(v,u) > \delta$ then the node u gets split and $d(u)$ is set to zero. computation proceeds from the leaves toward the root.

for each of leaf nodes 7, 8, 5, 9, 10 delay is zero.
The delay for any node is computed only after the delays for its children have been determined.

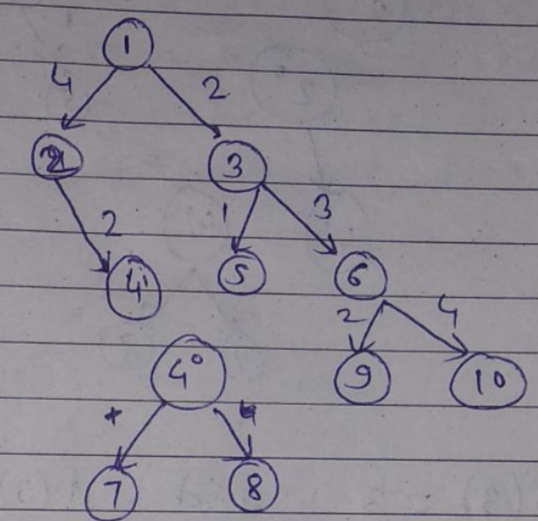
Let u be any node and $C(u)$ be the set of all children of u. Then $d(u)$ is given by

$$d(u) = \max_{v \in C(u)} \{ d(v) + w(u,v) \}$$

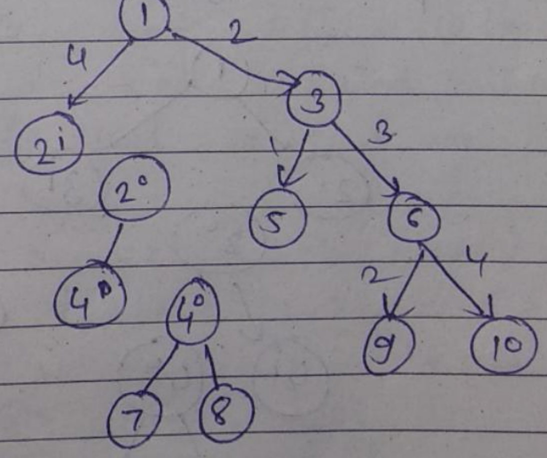
① for $\delta = 4$

using formula, for above tree

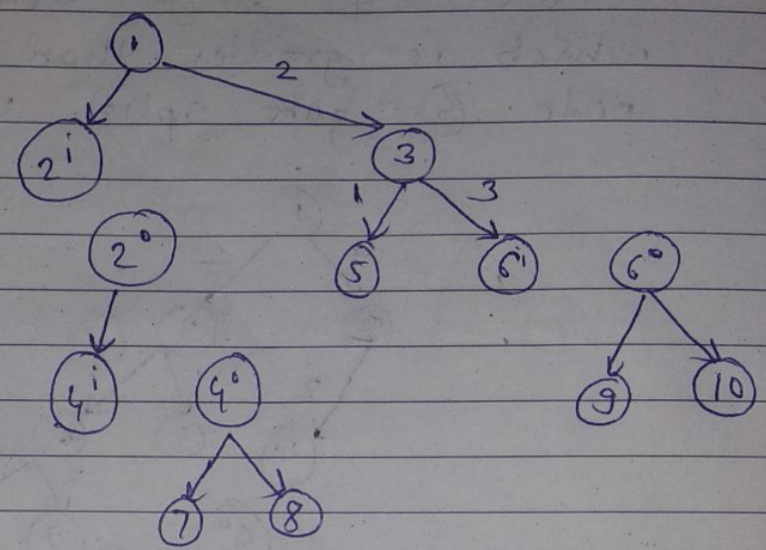
- $d(4) = 4$ since $d(4) + w(2,4) = 6$ which is greater than δ ie. 4 node ④ gets split. We set $d(4) = 0$.



- $d(2) = 2$, since $d(2) + w(1,2) = 2 + 4 = 6$ which is greater than δ node ② gets split and $d(2) = 0$.



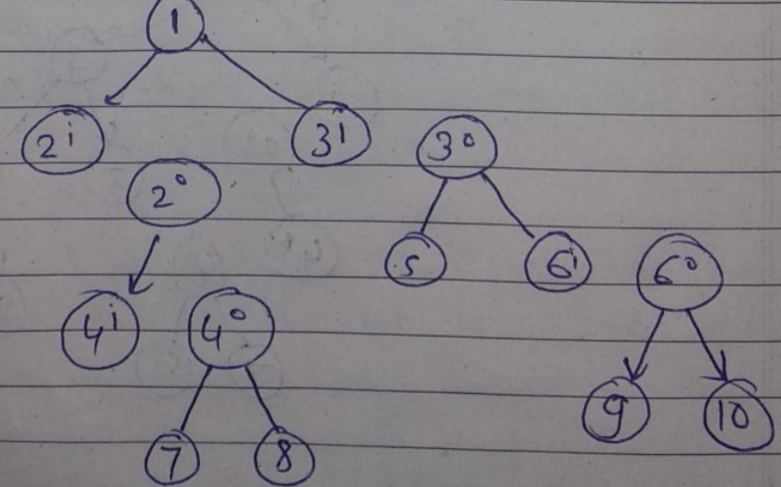
- $d(6) = 3$, since $d(6) + w(3,6)$ $= 3 + 3 = 6 > \delta$
node 6 split and we set $d(6) = 0$.



- $d(3) = 3$ and $d(3) + w(1,3) = 3+2=5$ which is ~~not~~ greater than δ, so node 3 ~~not~~ split. & set $d(3)=0$.

~~• d(1) = 5 which is less~~

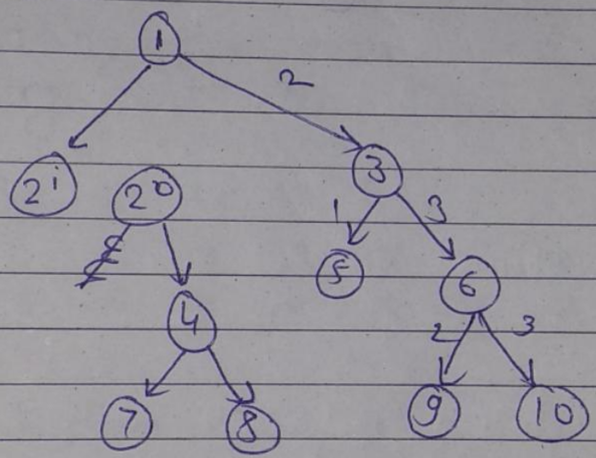
finally we get



b) for $\delta = 6$

- $d(4) = 4$, and $d(4) + \omega(2,4) = 6$ which is not greater than δ so node (4) not split and we set $d(4) = 4$

- $d(2) = 6$ ($\because d(4) + \omega(2,4) = 4 + 2 = 6$) since, $d(2) + \omega(1,2) = 6 + 4 = 10$ which is greater than δ so node (2) split and we set $d(2) = 0$.



- $d(6) = 3$ and $d(6) + \omega(3,6) = 6$ which is not greater than δ so node (6) not split & we set $d(6) = 3$

- $d(3) = 6$ as $d(6) + \omega(3,6) = 6$. since $d(3) + \omega(1,2) = 6 + 2 = 8$ which is greater than δ so node (3) split and we set $d(3) = 0$

we finally get,