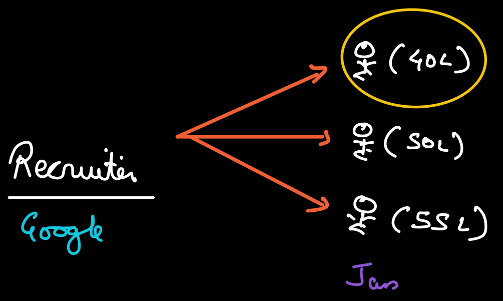
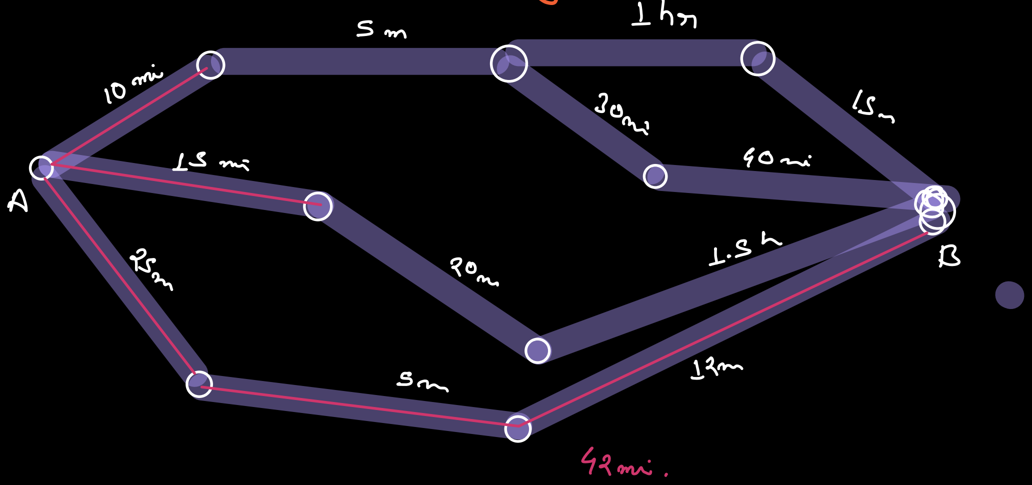


40L
56L
67L
35L

Feb

41L
36L
38L
35L

Greedy



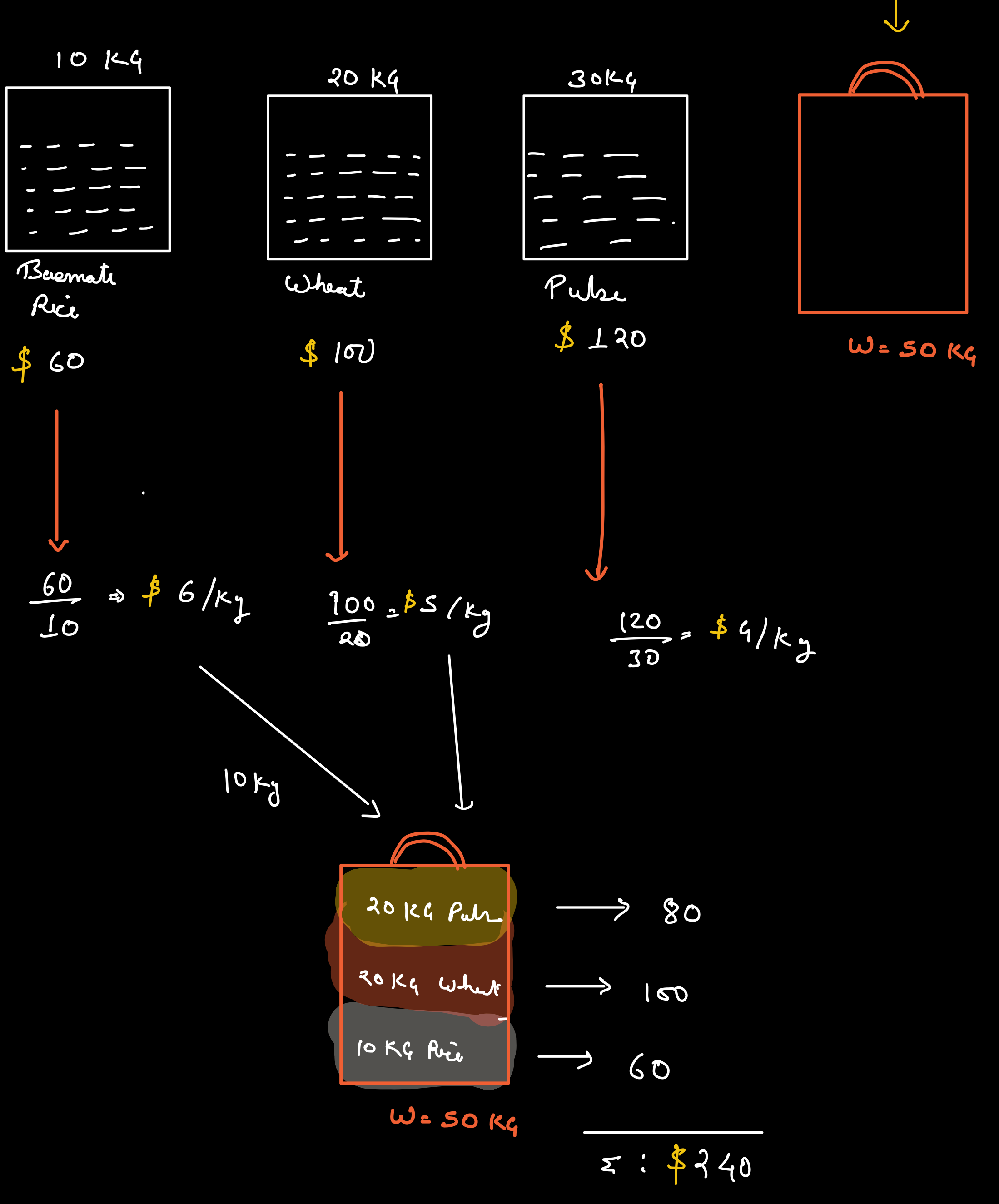
Q. Fractional Knapsack

Given N items < total weight / Cost

Given a bag of W kg.

Pick items to maximize the value of the bag.

Knapsack
Sack



Step I Sort all items based on per unit cost (val/wt ratio)

Step II Iterate, pick & keep updating bag's capacity.

0/1 Knapsack ⇒ DP

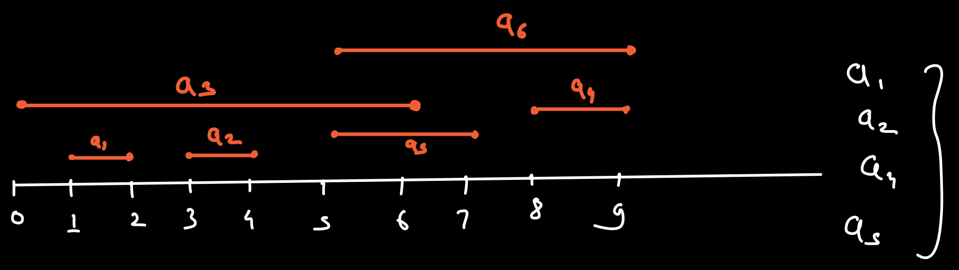
TC : O(N log N)

Amazon

Q Activity Selection

Given N activities with their start & end time. At a time only one activity can be performed. Maximize the no. of activities that you can perform.

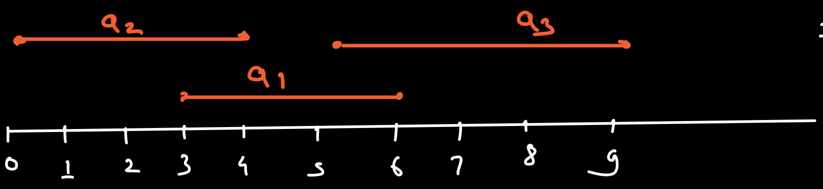
	a_1	a_2	a_3	a_4	a_5	a_6
Start	1	3	0	8	5	5
end	2	4	6	9	7	9



a_1 , a_2 , a_5 , a_4

Ex

	a_1	a_2	a_3
s	3	0	5
c	6	4	9



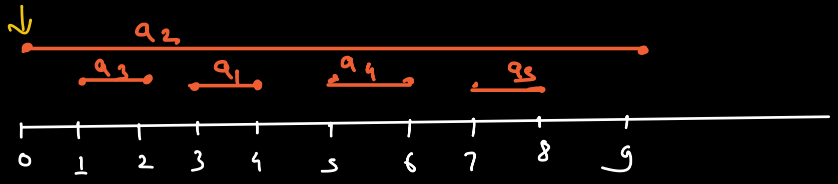
$\Rightarrow \underline{\underline{2}}$

a_2, a_3

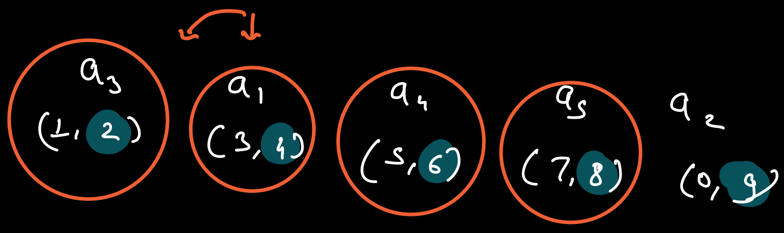
App 1   Sort based on duration ✗ ↑

App2

	a_1	a_2	a_3	a_4	a_5	
s	3	0	1	5	7	...
c	4	9	2	6	8	



$a_3, a_1, a_4, \underline{a_5}$ $\Rightarrow 4$



Amazon
Facebook

Q Job scheduling problem

Given N Jobs. → Deadline
→ Reward

* You need 1 day to complete one job

* On a given day, you can do a single job.

* A job can be performed on any day on or before the day of deadline.

Maximize the amount of reward.

Jobs	Deadlines	Reward
a	3	100
b	1	19
c	2	27
d	1	25
e	3	30

d	c	a							----
1	2	3	4	5	6	7	8	9	-.

⇒ 25 + 27 + 100 ⇒ 152

days

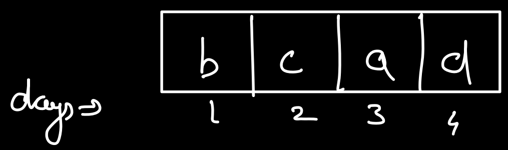
d	e	a							----
1	2	3	4	5	6	7	8	9	-.

⇒ 25 + 30 + 100 ⇒ 155

a	c	e							----
1	2	3	4	5	6	7	8	9	-.

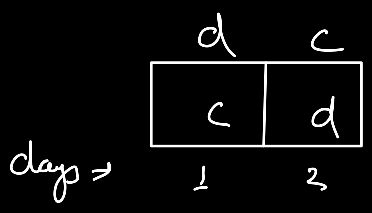
⇒ 100 + 27 + 30 ⇒ 157

Job	deadline	reward
a	3	100
b	1	100
c	2	100
d	4	100



→ Try to do the job ON the day of deadline.

Job	deadline	reward
a	2	200
b	2	100
c	2	500
d	2	300



⇒ 500 + 300

↑ Prioritize the job with highest reward.

job	deadline	reward
a	1	150
b	1	100
c	2	50

a	c
1	2

$\Rightarrow 150 + 50 \Rightarrow \underline{\underline{200}}$

job	deadline	reward
a	3	100
b	3	30
c	2	27
d	1	25
e	1	19

	c	b	a
day:	1	2	3

job	deadline	reward
a	3	6
b	1	5
c	3	3
d	2	8
e	3	7

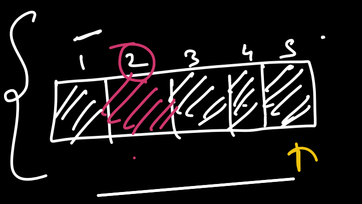
a	d	e
1	2	3

Step I — Sort based on reward in DESC order.

Step II — Iterate over the sorted data;

for every job
iterate on the day array
from deadline day to day 1 (←)
& do the job in the first available slot.

TreeSet



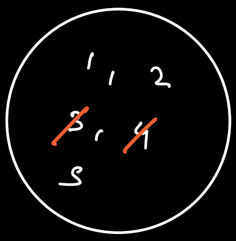
N → jobs

Max deadline → d

TC : $O(N \log N) + \underline{O(N d)} \rightarrow O(N \log d)$

For every job,

we want the max available day slot which is less than or equal to the deadline.



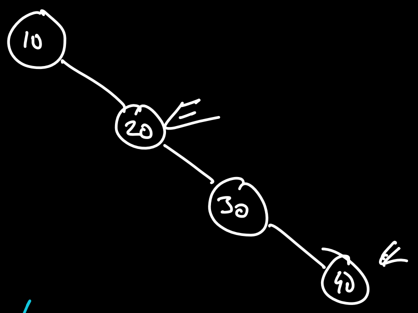
a → 4

b → 3

c → 4

Binary Search Tree

O(N)

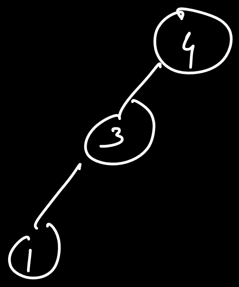


a → 2

b → 5

c → 7

d → 6



AVL / R-B tree

TreeSet / Set
(C++)