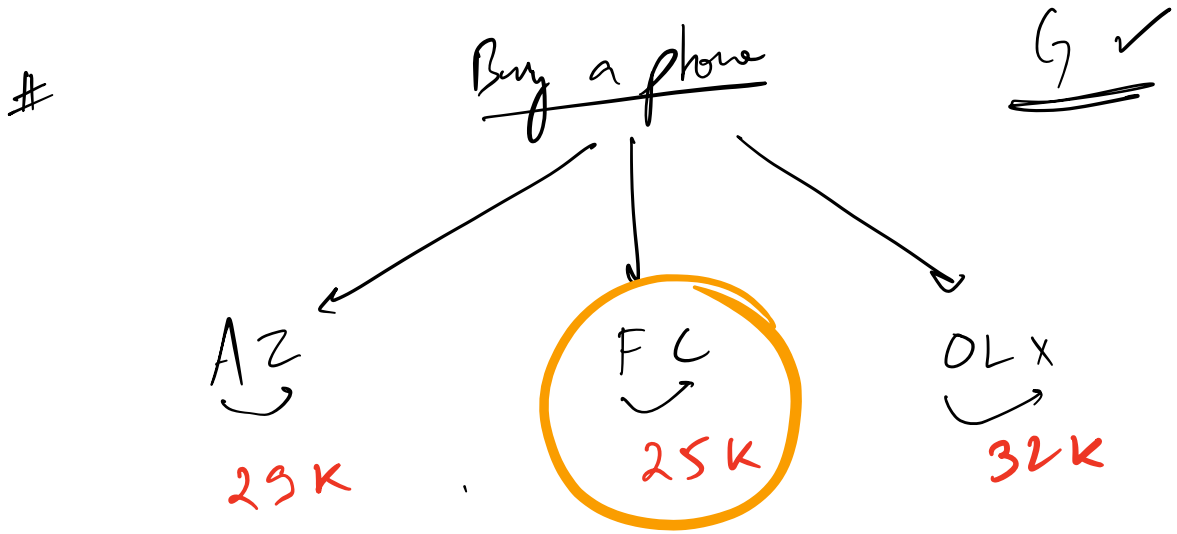
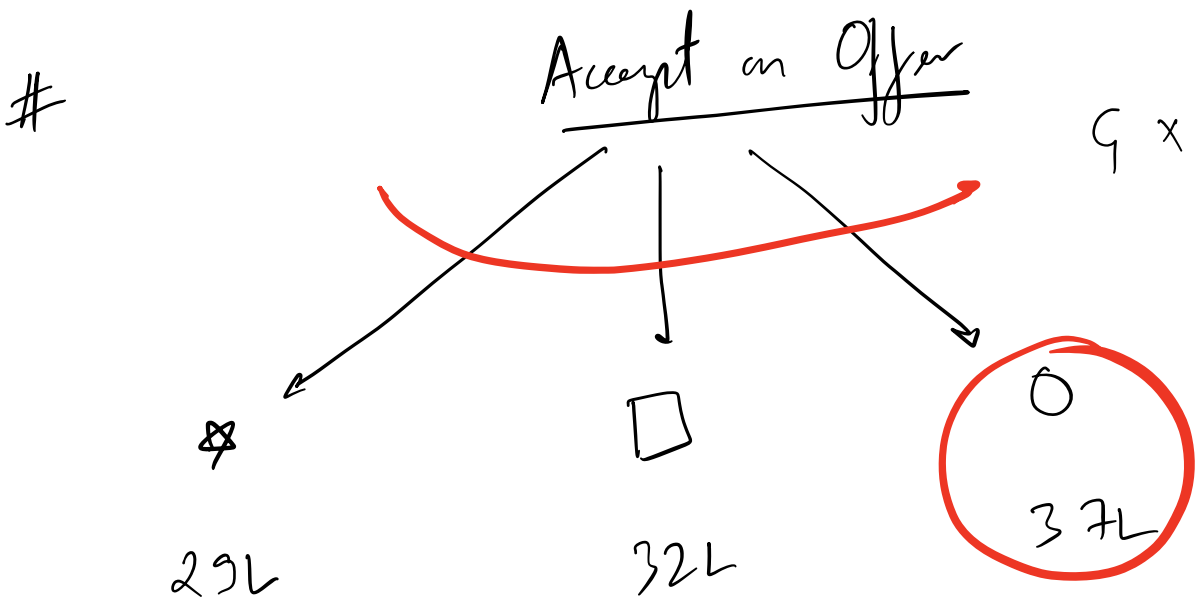
Greedy Algo

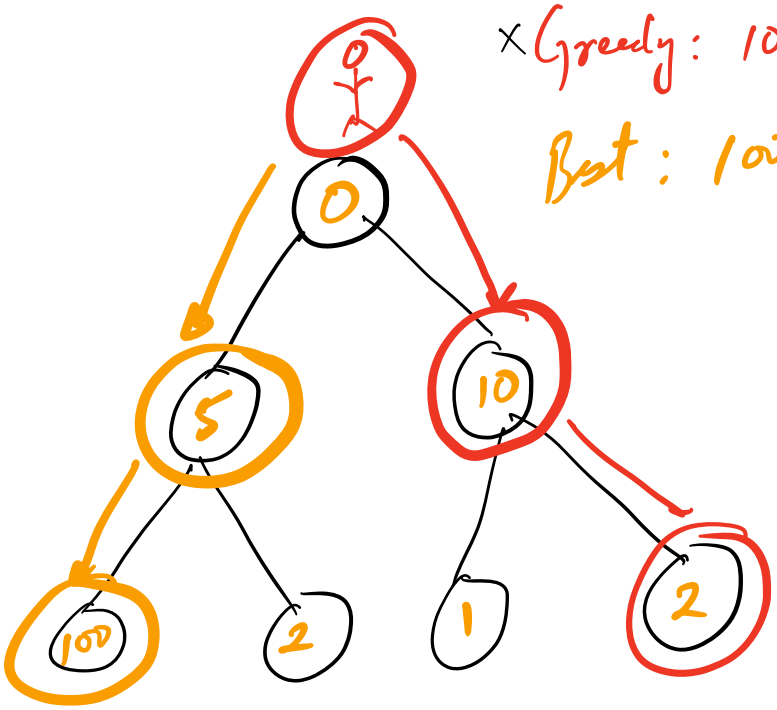




Stocks?
WLB?
Benefits?

0

→ Greedy strategy might not always lead to the MOST optimal sol".



× Greedy: 10+2 = 12

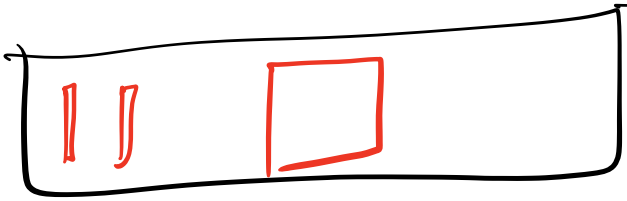
Best: 100+5 = 105

#

#

Read: 20 books

100 book

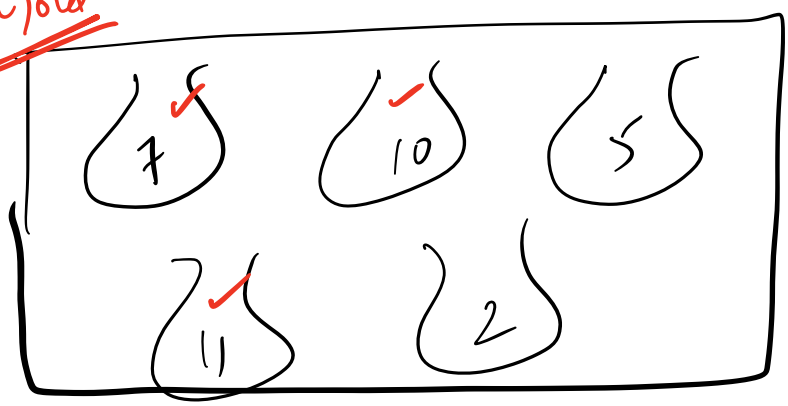


pick the thinnest 20 books.

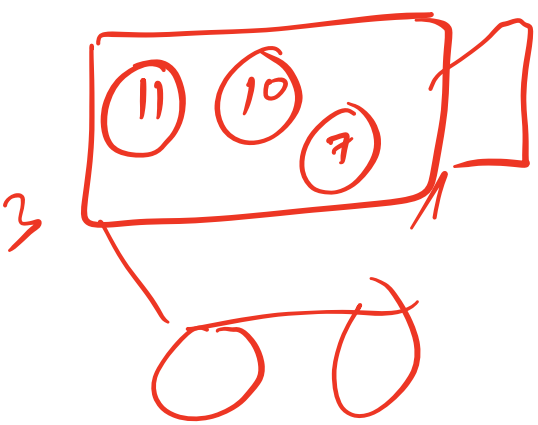
Ropes (Heap) G ✓

pick the smallest 2 ropes → tie

Gold G ✓



pick the largest 3



---

④ Greedy: Think greedily at every step w/o considering future / all possibilities.

④ Some problems which can be solved greedily others cannot be!

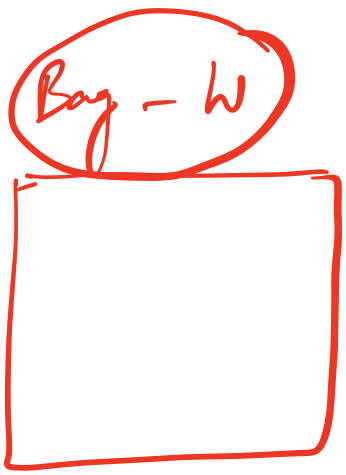
---

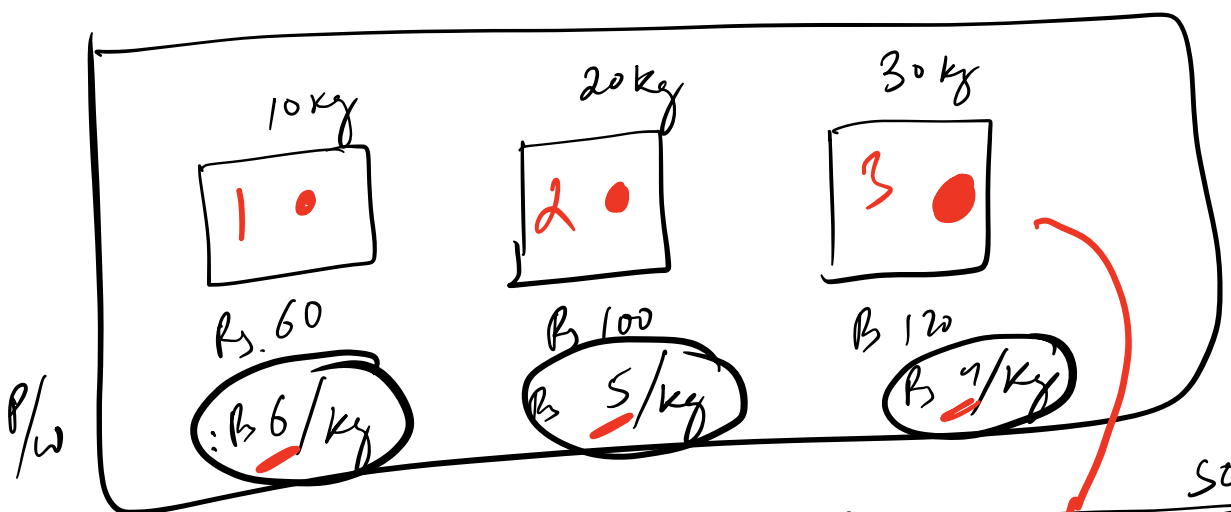
1) Fractional Knapsack

break items

N items → Weight

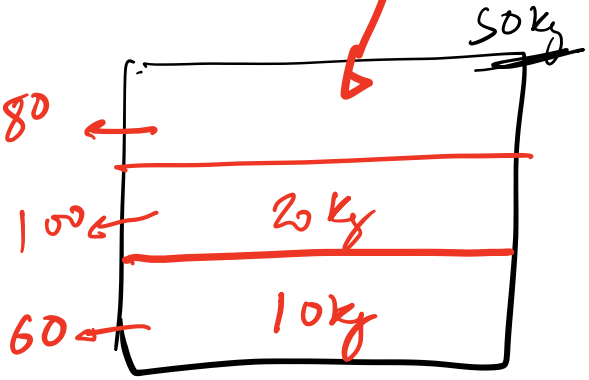
N items → Value



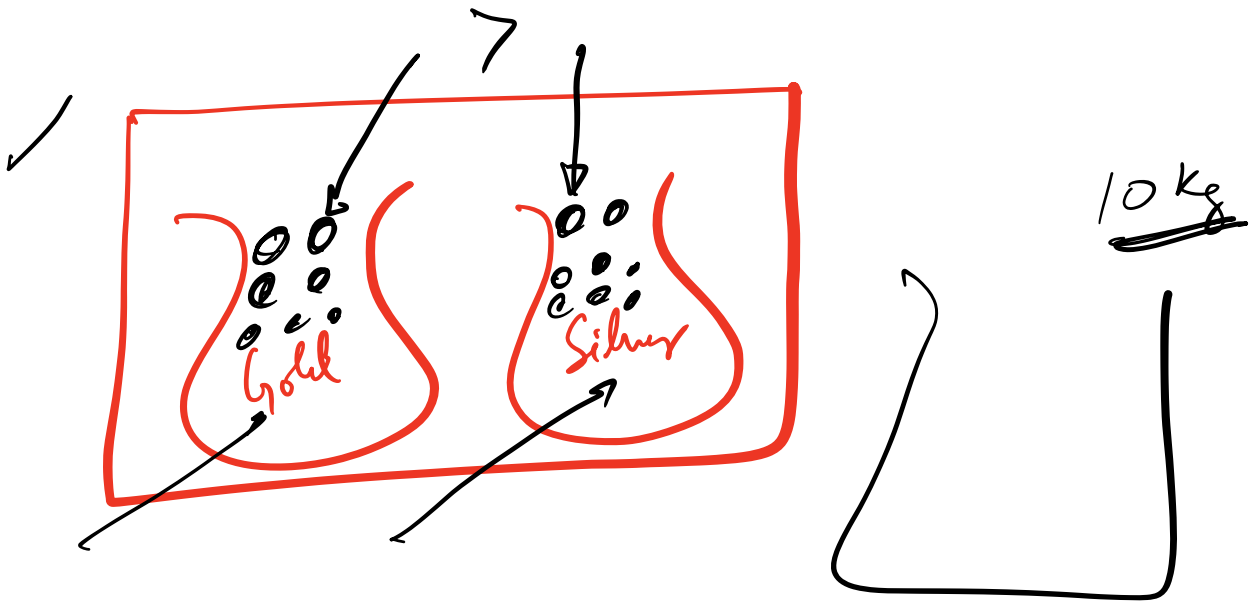


$\frac{20}{30} \times 120$

20 (80



80 + 100 + 60 = 240

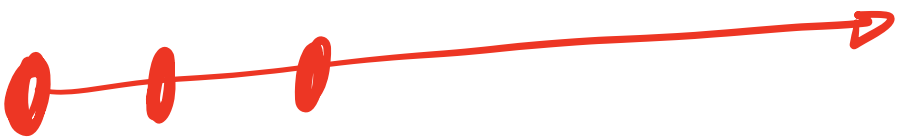


Strategy:

find the P/wt of every item.

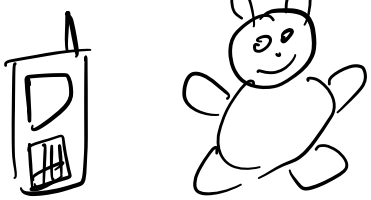
Sort them in the DEC order of P/wt.

MAX



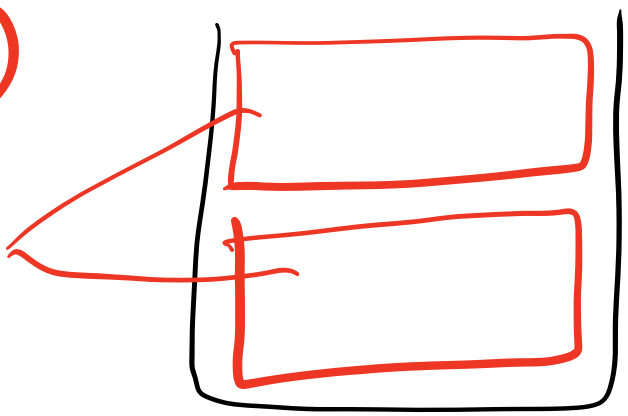
0/1 Knapsack → DP

Cannot break items!



1) MAX Val X

Greedy X ✓



Activity Selection Problem

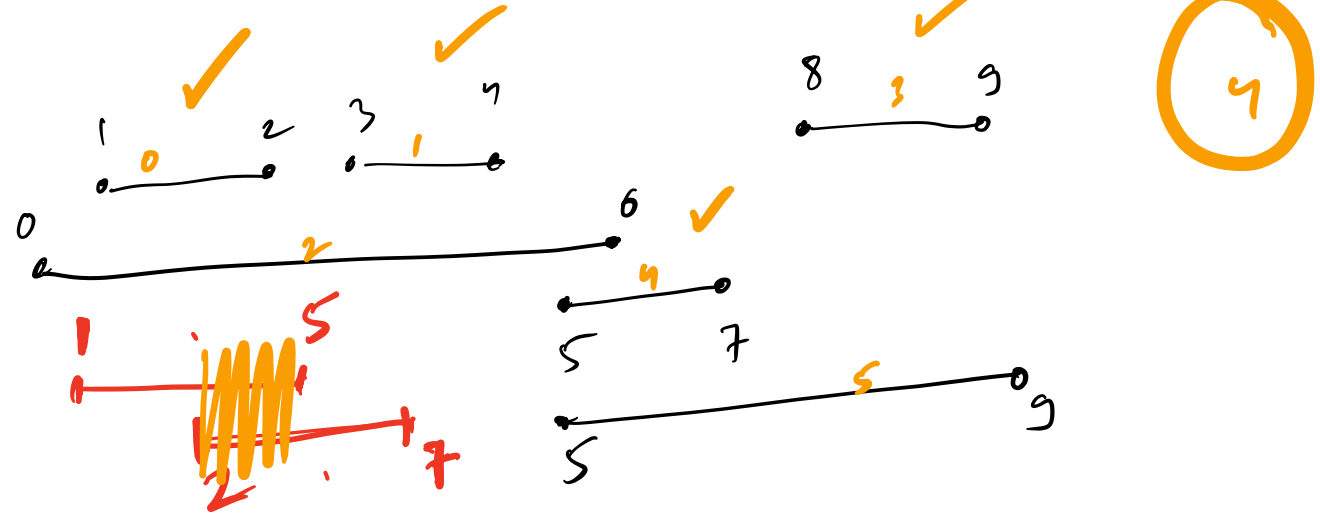
Given N activities.

[S , E]

① At 1 time you will be performing only 1 Act!

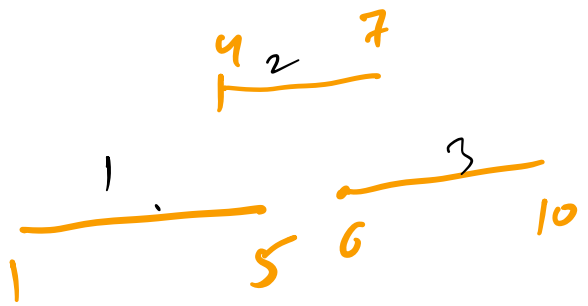
② Find the MAX no. of activities that you can perform.

	0	1	2	3	4	5
S	1	3	0	8	5	5
E	2	4	6	9	7	9



1) Pick the act. having the least duration first.

X

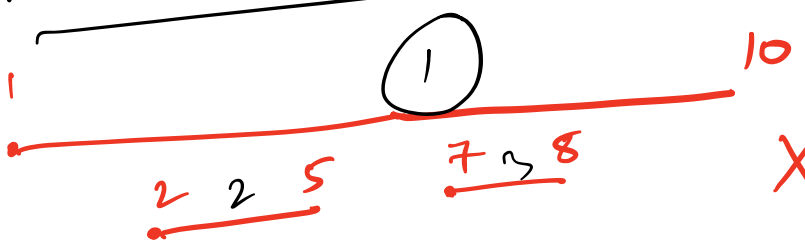


G → 2 : 1

S → 1, 3 : 2

2) Pick the act. with the smallest S.T first.

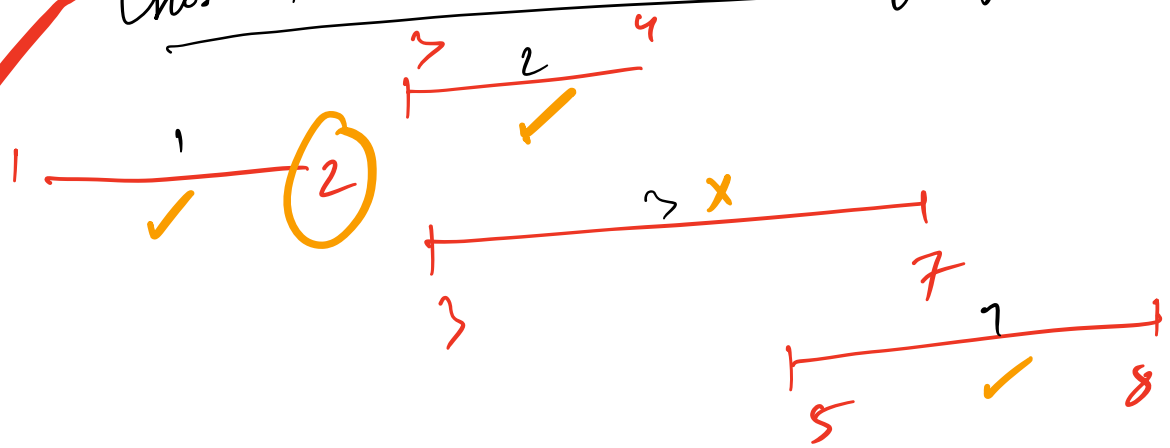
X

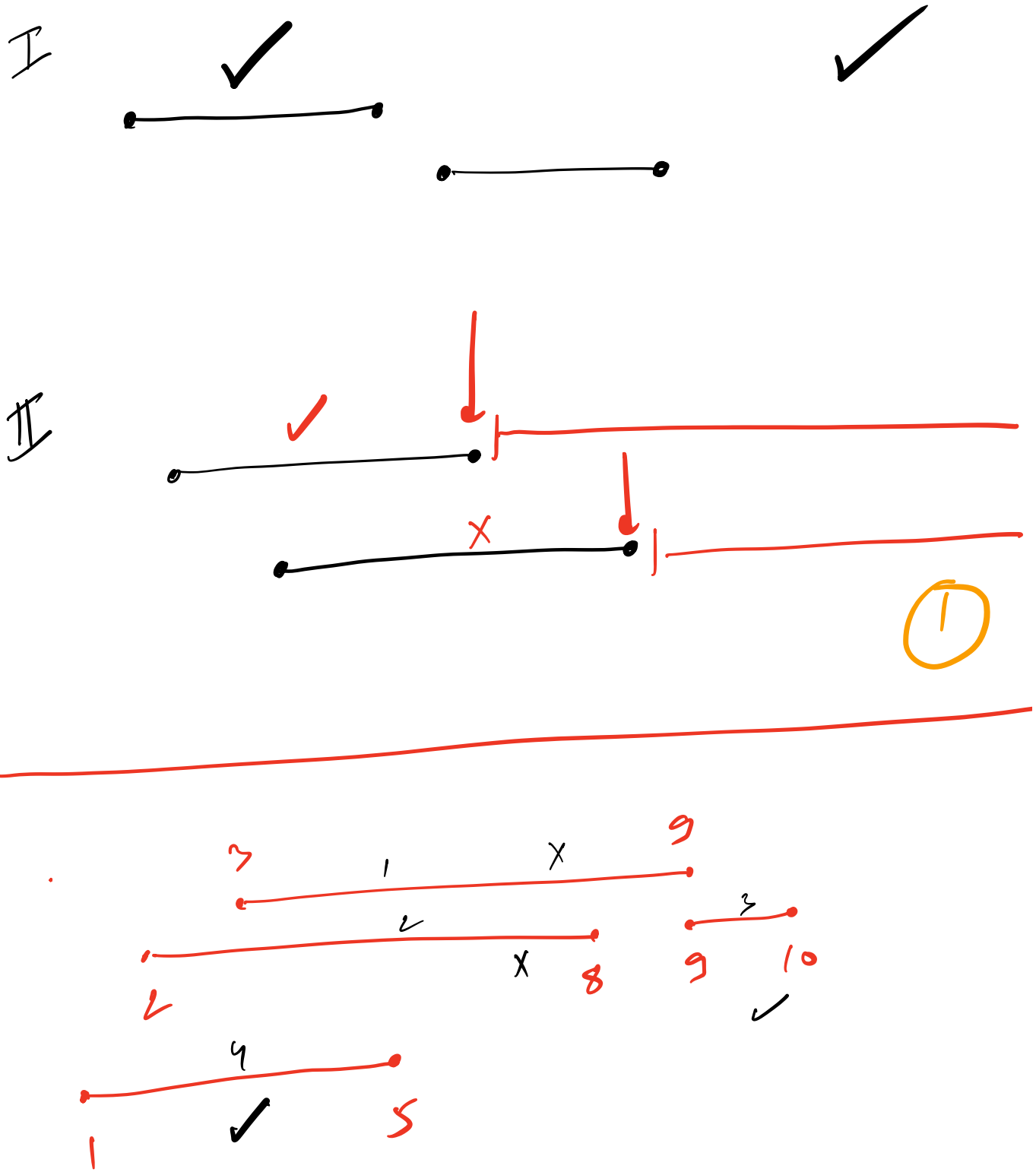


X G : 1

S : 2, 3

3) Chose the ACT which is ending first.

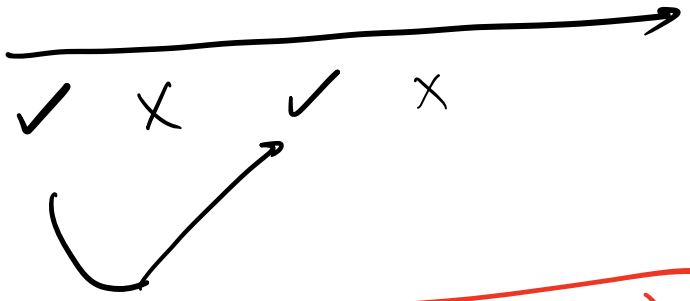



I
II
X
1
3
1
X
9
2
X
8
3
9
10
4
1
5

1) Sort them acc to INC E.T.

NlgN

2)



N

TC: O(NlgN)

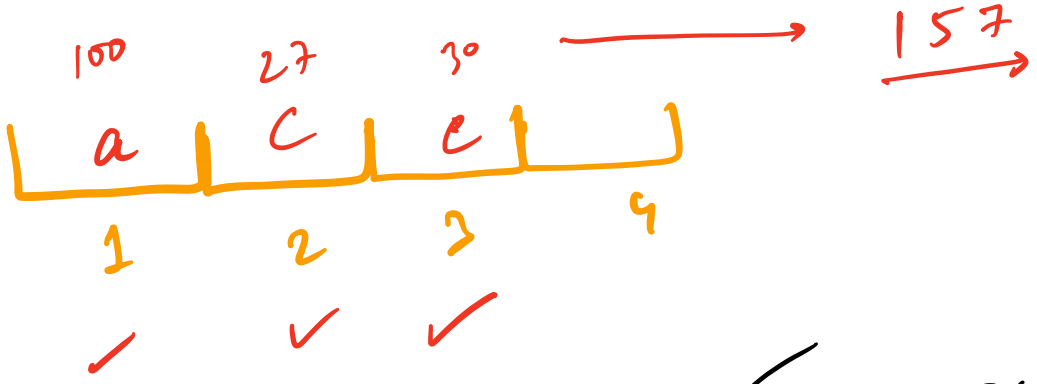
Q JOB SCHEDULING

Given N jobs → reward ✓
→ deadline.

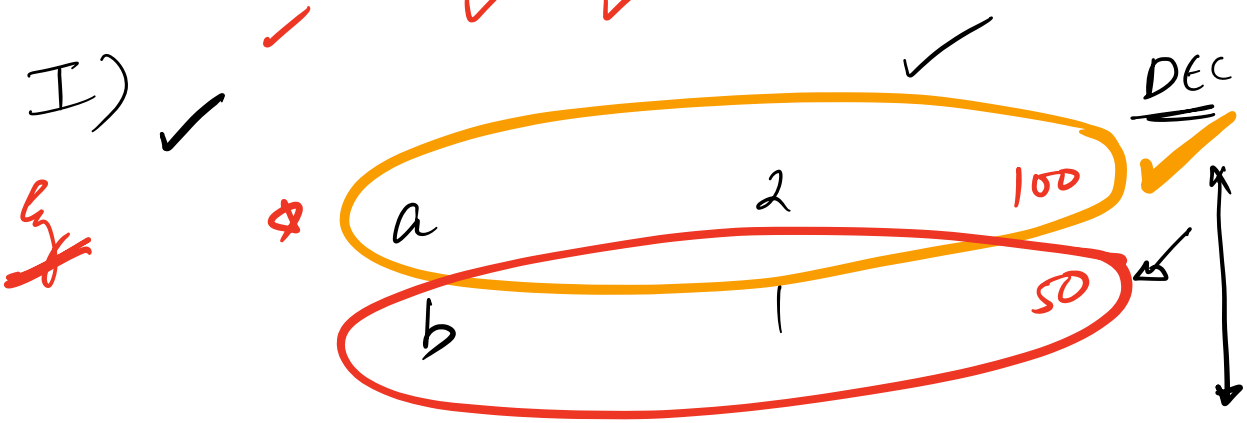
1) At a time, a single job can be performed
2) A job takes 1 unit of time to complete

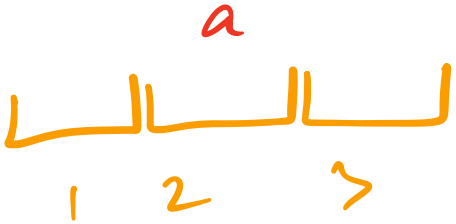
MAX the amount of reward!

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



I)





Jobs	deadline	reward	
a	3	✓ 100	1
b	1	✗ 19	5
c	2 → 1, 2	✓ 27	3
d	1	✗ 25	4 ←
e	3	✓ 30	2

00 27 30 →

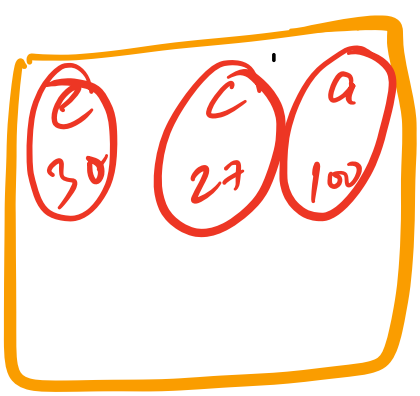
II Sort INC DEADLINE →

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

100 27 30 →

Sort

J	b	d	c	a	e
d:	1	1	2	3	3
r:	19	25	27	100	30



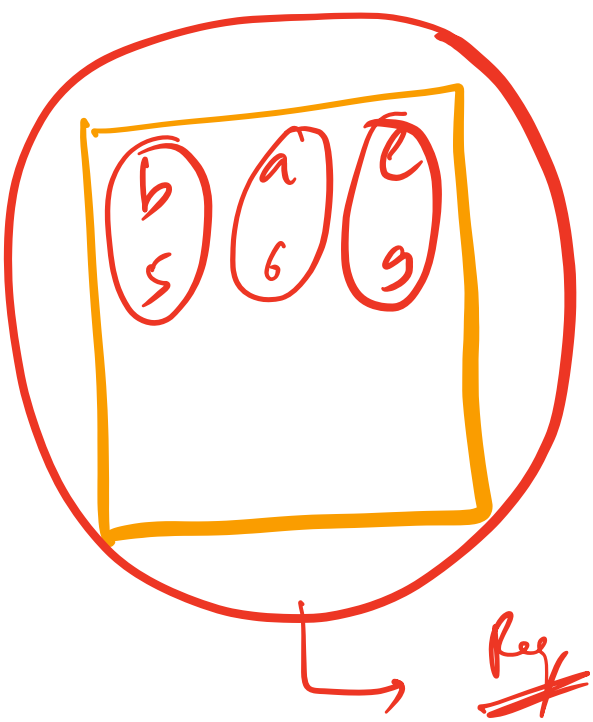
day = 3

	a	b	c	d	e
d:	3	1	3	2	3
r:	6	5	3	1	9

Sort → NlgN

	b	d	e	c	a
d:	1	2	3	3	3
r:	5	1	9	3	6

NlgN



d = 3

5+6+9 !

Req
1) Min element. ← O(1)
2) Del. Min ← O(lgn)
3) Insert ← O(lgn)

MIN Heap

TC: O(N log N)

SC: O(N)

```
days = 0;
MH <> ;
Sort (jobs) ; // INC dead.

f( i=0; i< N; i++) {
    if( deadline[i] > days ) {
        MH. insert( reward[i] );
        days++;
    }
    else {
        min = MH. getMin();
        if( reward[i] > min ){
```

```
        mn.deltop();
        mn.insert(second(i));
    }
  }
}
```

---

1 Distribute Candies →

N students. → MARKS.

1) Every student should get atleast 1 candy
2) a student should get more candies than it's neighbour if his/her marks are higher than the neighbour's marks.

GOAL: DISTRIBUTE least no. of candies.

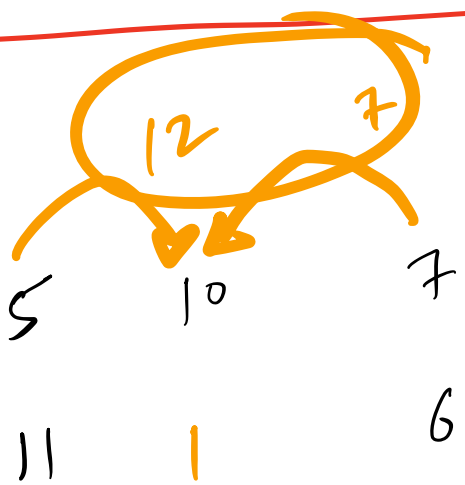
ex:

A: 1 5 2 1 6

C: 1 3 2 1 2

A: 2 6 3 1 10 12 20 5 2

C: 1 3 2 1 2 3 4 2 1



A: 2 6 3 1 10 12 20 5 2

L: 1 2 1 1 2 3 4 1 1

R: 1 3 2 1 1 1 3 2 1

C: 1 3 2 1 2 3 4 2 1

TC: O(N)

SC: O(N)