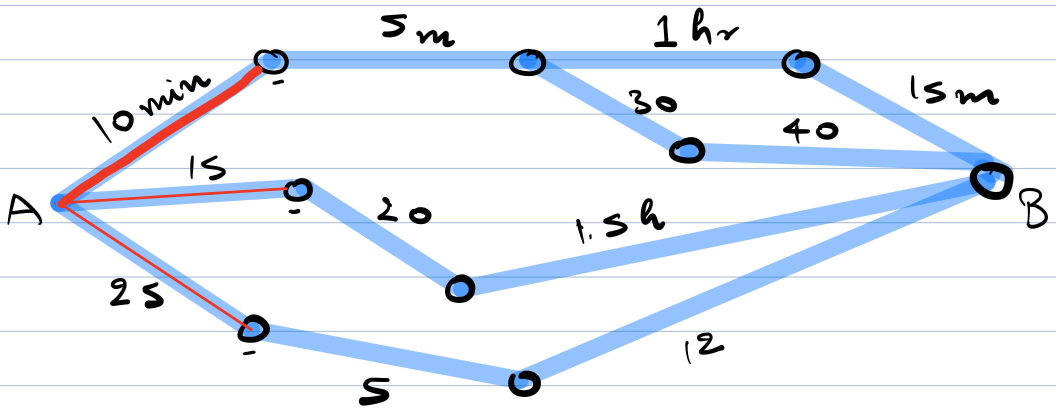
Recruiter
Google

Jan	Feb	Mar
[✓ 웃 (40L)]	웃 (40L)	웃 (41L)
웃 (50L)	웃 (56L)	웃 (36L)
웃 (55L)	웃 (67L)	웃 (38L)
	[✓ 웃 (35L)]	[✓ 웃 (35L)]

Reduce overall cost



Goal : Reach B in min time

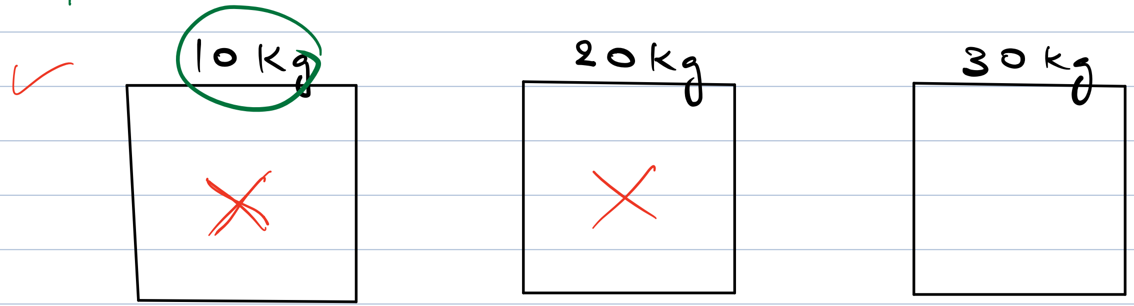
Q Fractional Knapsack

Given N items → total weight
→ cost (value)

Given a bag of capacity w

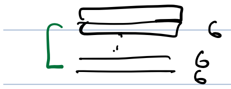
Pick some items st. value in the bag is maximized.

Supermarket



Rice	Wheat	Pulse
$ 60	$ 100	$ 120
1 Kg = $6	$ 5	$ 4

30 kg pulse = 120
20 kg wheat = 100
220 ✗



20 kg pulse	→ $ 80
20 kg wheat	→ $ 100
10 Kg Rice	→ $ 60
w = 50K	$ 240

?

Constraint : Either select all or select nothing

20	→
20 kg wheat	→ 100
10 Kg Rice	→ 60
w = 50K	160

0/1 Knapsack ⇒ DP

Steps

1) Sort all the items on the basis of per unit cost. (total val / total weight)

2) Iterate, pick the best & keep updating the bag's capacity.

$$T.C. = O(N \log N + N)$$
$$= O(N \log N)$$

Q Gautam (CEO) [Famous]

BMW approaches Gautam Special sale

N cars

A → A[i] time till which the ith car is free of cost (sale is active

B → B[i] beauty of the ith car

Maximise the overall beauty value of the purchased car.

→ It takes 1 unit of time to purchase one car

0 1 2 3 4

A → [3, 1, 3, 2, 3]

B → [6, 5, 3, 1, 9]

× × × × ×

T = 0   9   5

T = 1   6   6

T = 2   3   9

T = 3   18   20

1) Sort the cars on the basis of sale end time.

A → 1, 5, 5, 5, 3, 3, 3

B → 5, 4, 3, 8, 20, 7, 10

→ 0   1   2   3   4   5   6

A → 1 , 3 , 3 , 3 , 5 , 5 , 5

B → 5 , 20 , 7 , 10 , 4 , 3 , 8

T = 0   ~~0 (5)~~   3 (10)

T = 1   1 (20)

T = 2   2 (7)

T = 3   4 (4)

T = 4   ~~5 (3)~~   6 (8)

T = 5

min ⇒ Min Heap

Code

```
int t = 0;
```

```
minHeap   m ;
total Beauty = 0;
for (i = 0; i < N; i ++) {
        if ( t < A[i] ) {
                total Beauty += B[i];
                m. insert ( B[i] );
                t ++;
        }
        else {
                if ( B[i] > m. peek() ) {
                        totalBeauty -= m. deleteMin();
                        m. insert ( B[i] );
                        total Beauty += B[i];
                }
        }
}
```

Iterate over the heap & find total beauty sum.

$$T.C. = O(N\log N + N\log N)$$

$$= O(N\log N)$$

$$S.C. = O(N)$$

Q Given N jobs that we need to perform.

Paypal

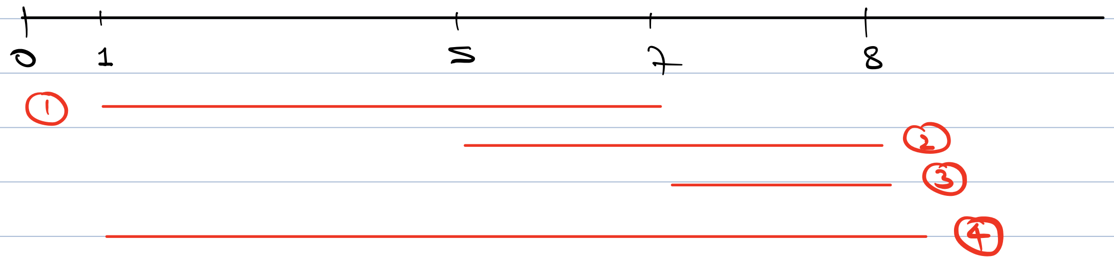
Each job → Start time
→ End time

Const : Only able to perform 1 job at a time

Find the max no. of jobs that we can perform.

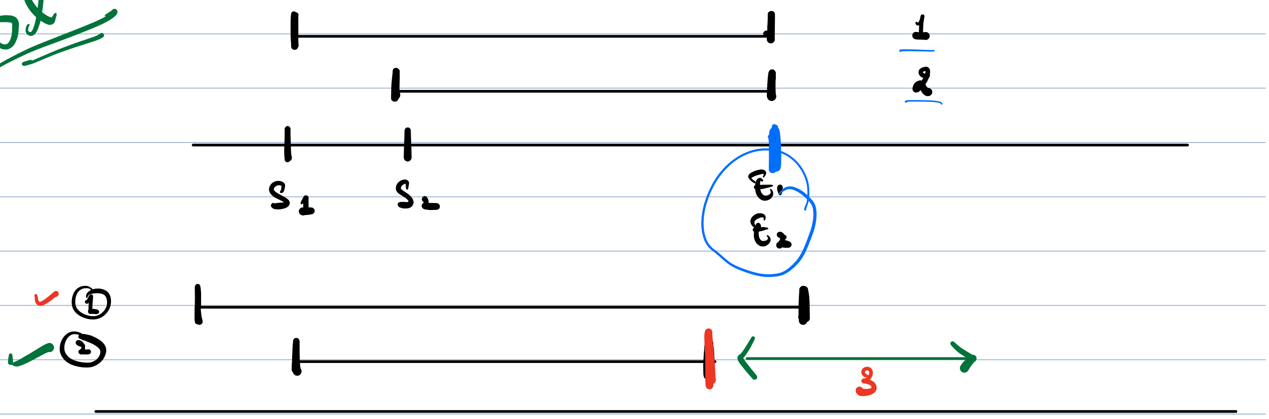
	0	1	2	3
Start	1	5	7	1
End	7	8	8	8

0 . & 2   Ans = 2

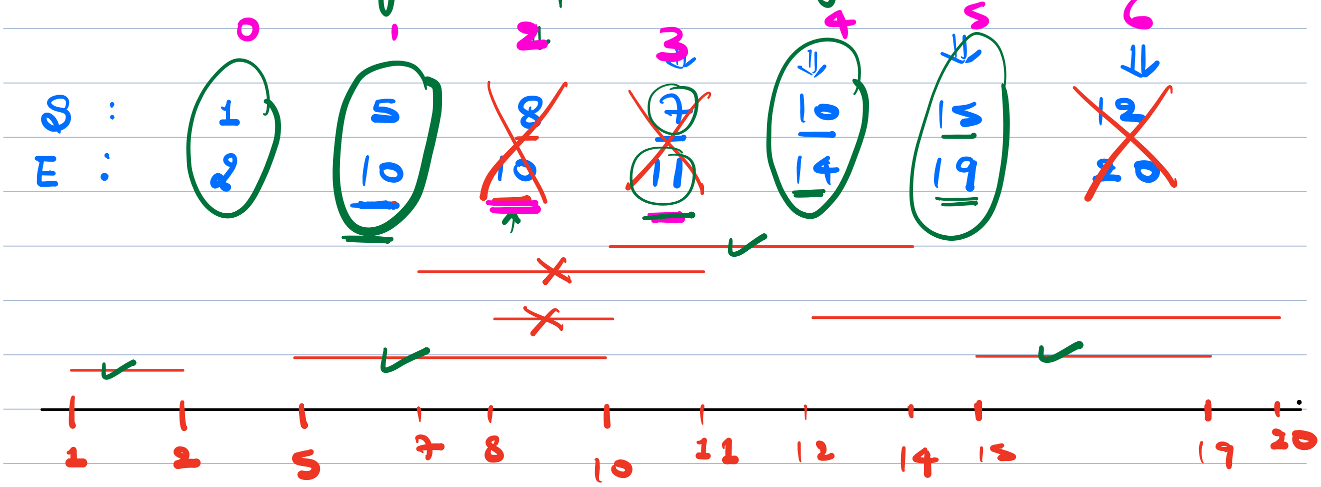


Soln

min 2 start times



Goal : Every time select the job with the least end time. out of the possible jobs



Assume jobs are sorted on basis of ET

Code

```
count = 1;
lastEndTime = E[0];

for (i=1; i<N; i++) {
        if (S[i] >= lastEndTime) {
                count++;
                lastEndTime = E[i];
        }
}
```

$$T.C. = O(N\log N + N)$$

$$= O(N\log N)$$

S.C. = O(N)

Doubt

~~1 5 6~~ Invalid

~~3 1 2~~

~~5 6 1~~

~~1 2 3~~