

Welcome 😊

Agenda: Greedy algo.
3 questions.

Greedy Algo. (Be greedy but be cautious)

Greedy \rightarrow max profits / min losses.

iPhone (amazon) \rightarrow 1.3 lac

iPhone (flipkart) \rightarrow 1.2 lac

Q Flipkart Grocery Services.

A \rightarrow expiration time of each item.

B \rightarrow profit gained by each item.

Maximize the profits.

A: [⁰3 ¹1 ²3 ³2 ⁴3]

B: [6 5 3 1 9]

Time index price.

1 4 9

2 0 6

3 1 5 X expired

3 2 3

4 3 1 X expired

Total profit = 18

Time	order	price
1	1	5
2	4	9
3	0	6
		<u>20</u>
		$> \underline{\underline{18}}$

Qvig [1 2]
[3 1500]

1500 ✓ ✗
 $3 + 1500 \Rightarrow \underline{\underline{1503}}$ ✓

Approach

eg: [⁰1 ¹3 ²3 ³3 ⁴5 ⁵5 ⁶5 ⁷8]
[5 2 7 1 4 3 8 1]

Time	Order	Price.
1	0	5
2	1	2
2	2	7
3	4	4
4	5	3
5	6	8
6	7	1
		$\Rightarrow \underline{\underline{28}}$

MinHeap D-S

eg: $\begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \text{[} & 1 & 3 & 3 & 3 & 5 & 5 & 5 & 8 \end{matrix}$

$\begin{matrix} & & & & & & \downarrow \\ \text{[} & 5 & 2 & 7 & 1 & 4 & 3 & 8 & 1 \end{matrix}$

index

$T=1 \rightarrow 0$

$\cdot 2 \rightarrow \cdot 1 \cdot$

$2 \rightarrow 2$

$4 \rightarrow 3 \times$

$3 \rightarrow 4$

$4 \rightarrow 5$

$5 \rightarrow 6$

$6 \rightarrow 7$

code

1. Sort w.r.t time

2. $t=0$

for ($i=0 \rightarrow N-1$)

{ if ($A[i] > t$) // not expired

{ minheap.insert ($B[i]$)

$t++$

}

else // expired

{

if ($B[i] \leq \text{root of heap}$)
// ignore.

else

extractMin ()

heapMin.insert ($B[i]$)

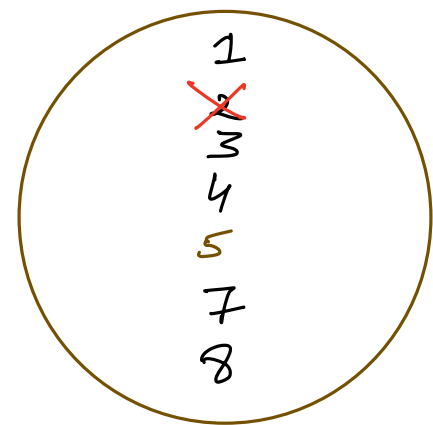
}

$\Rightarrow \text{Sum} = \underline{\underline{28}}$

3.

Remove all elements from heap and return their sum.

T.C $\Rightarrow O(N \log N)$
S.C $\Rightarrow O(N)$



Candy Distribution

There are N children with marks. Teacher has to give candies to them.

- a) Every child must have at least 1 candy.
 - b) Children with higher marks than its neighbours must have more candies than its neighbour.
 - c) No constraints/condition for equal marks.
- Find min. no. of candies to do this.

eg: $A: [1 \quad 5 \quad 2 \quad 1]$

$C \rightarrow$

1	1	2	1
	2	2	
	3		

Quiz $A: [4 \quad 4 \quad 4 \quad 4 \quad 4]$

$C \rightarrow$

1	1	1	1	1
---	---	---	---	---

\Rightarrow 5

eg: $A: [6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1]$

$C \rightarrow$

1	1	1	1	2	1
6	5	4	3	2	

1) $\forall i \Rightarrow C_i \geq 1$

2) if $(A[i] > A[i-1])$ // left to right
 $C[i] = C[i-1] + 1$

\Rightarrow if $(A[i] > A[i+1])$ // right to left.
 $c[i] = \max(c[i], c[i+1] + 1)$

eg: A: [1 6 3 1 10 12 20 5 2]
 c 1 ~~2~~ ~~1~~ 1 ~~2~~ ~~3~~ 1 ~~2~~ 1
 3 2 3 4 2

Code $\forall i \Rightarrow c_i \geq 1$

```

for (i → 1 to N-1)
{
    if (A[i] > A[i-1]) // left to right
        c[i] = c[i-1] + 1
}
ans = A[N-1]
for (i → N-2 to 0)
{
    if (A[i] > A[i+1]) // right to left.
        c[i] = max(c[i], c[i+1] + 1)
    ans += c[i]
}

return ans
  
```

T.C $\rightarrow O(N)$

S.C $\rightarrow O(N)$

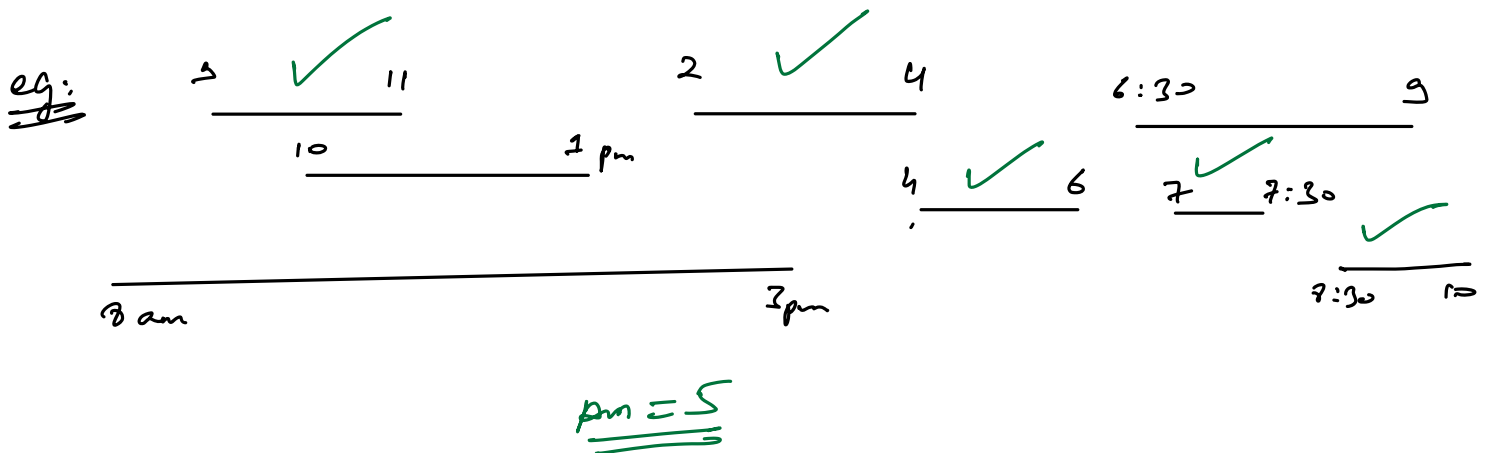
10:33 →

↓ ←

✓✓

Given N jobs with start & end time.

Find max. jobs that can be completed if only one job can be done at a time.

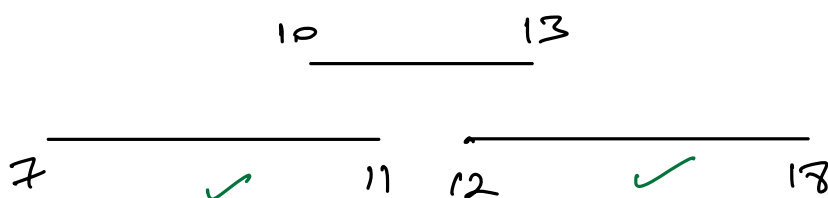


Soln

Start : [1 5 8 7 12 13]

End : [2 10 10 11 20 19]

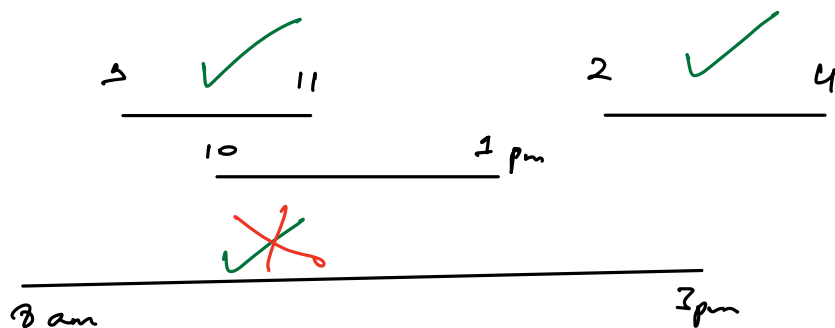
Case 1 least duration jobs.



Ans \Rightarrow ~~1~~ 2

Case 2

Greedy by start time.



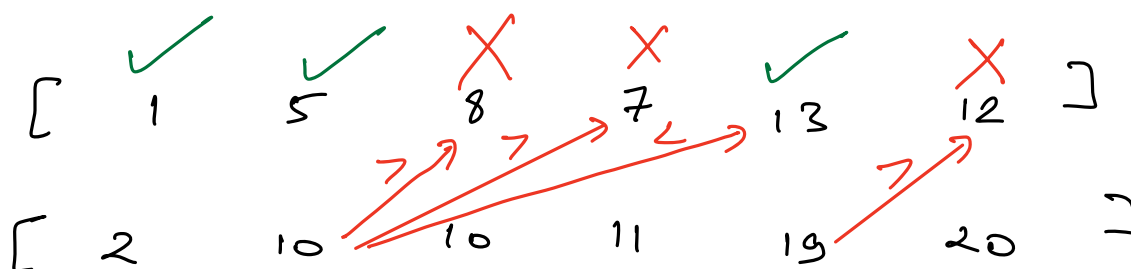
Ans = ~~1~~ 2

Case 3

Greedy on end time.

eg: [1 5 8 7 12 13]
 [2 10 10 11 20 19]

Sort based on end time.



Ans = 3

Code

1. Sort on the basis of end time.

2. ans = 1 lastEndTime = E[0].

for (i → 1 to N-1)

{ if (s[i] ≥ lastEndTime) {

 ans ++

 lastEndTime = E[i]

}
return ans

Contest

Trees Heap, Greedy