Greedy → max profit / min loss

Q → There is a **limited time sale** going on for toys,
A[i] → **sale end time** for i^{th} toy.
B[i] → Beauty of i^{th} toy.
Time starts from **T = 0** & it takes **1 unit of time** to **buy one toy** & toy can only be bought if **T < A[i]**.
Buy toys s.t **sum of beauty** of toys is **maximized**.

A = [3 1 3 2 3]
B = [6 5 3 1 9]
1 2 3 4 5

	toy		toy
T=0	5 → 9	T=0	2 → 5
✗	1 → 6	✗	5 → 9
2	3 → 3	✗	1 → 6
	18 ✗	3	20 ✓

1 2
A = [1 2]
B = [3 1500]
T = 0 1

Ans = 3 + 1500 = 1503 ✓

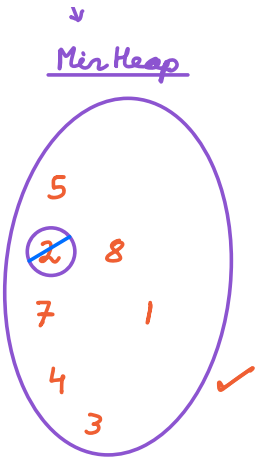
Greedy → Buy Everything. → ascending order of time. ✓

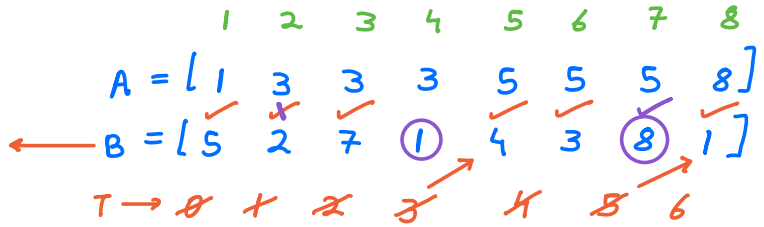
1 2 3 4 5 6 7 8
A = [1 3 3 3 5 5 5 8]
B = [5 2 7 1 4 3 8 1]
T → 0 1 2 3 4 5

Ans = 5 + 7 + 4 + 3 + 8 + 1
= 28 ✓

min B[i] selected if B[i] is very less → ignore.

Correcting or incorrect step from past.






```
if (T >= A[i]) {
    if (B[i] <= root of minHeap) → ignore
    else {
        getMin() from Heap.  // remove
        insert (B[i])
    }
} else {
    insert (B[i])
    T++
}
```

TC = O(N log(N))

SC = O(N)

Q → There are N students with marks.
Teacher has to give them candies s.t
a) Every student should have atleast one candy.
b) Student with more marks than neighbours (i-1, i+1) have more candies than them.
Find min candies to distribute.

A = [1 5 2 1] (1 2 3 4)

C → 1 3 2 1 Ans = 7

A = [4 4 4 4 4] (1 2 3 4 5)

C → 1 1 1 1 1 Ans = 5

A = [8 10 6 2] (1 2 3 4)

1 3 2 1 Ans = 7

	1	2	3	4	5	6	7	8	9
A =	1	6	3	1	10	12	20	5	2
C =	1	~~2~~ 3	2	1	2	3	4	2	1

Ans = 19

1) $\forall i$, $C[i] >= 1$ ✓

2) a) $\forall i$, if $(A[i] > A[i-1]) \Rightarrow C[i] > C[i-1]$ → $C[i] = C[i-1] + 1$ ✓ (left → right, greedy)

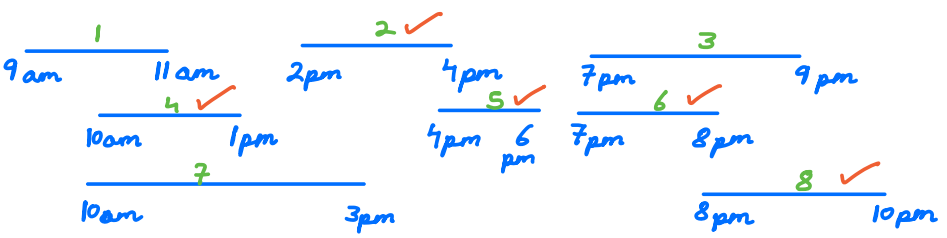
b) $\forall i$, if $(A[i] > A[i+1]) \Rightarrow C[i] > C[i+1]$ → if $(C[i] <= C[i+1])$ $C[i] = C[i+1] + 1$ ✓ (right to left)

TC = O(N) SC = ~~O(1)~~ O(N) ✓

10:35 PM

Flipkart Railway Interview

Q → Given N jobs with their start & end time. Find max jobs that can be completed if only one job can be done at a time. $S[i] >= E[i-1]$



Ans = 5

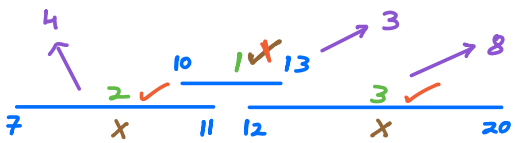
	1	2	3	4	5	6
S =	1	5	8	7	12	13
E =	2	10	10	11	20	19

1 1

Ans = 3

Greedy →

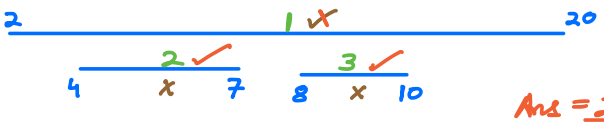
1) Duration → ✗



Ans = 2

2) Start Time → x

3) End Time → ✓ end early ⇒ more options to start. ✓



Ans = 2

start early + less duration ⇒ end early ✓

```
// sort wrt end time
ans = 1
lastEnd = E[0]
for i → 1 to (N-1) {
    if (S[i] >= lastEnd) {
        ans++
        lastEnd = E[i]
    }
}
return ans
```

TC = O(N log(N)) SC = O(1)

Q → N people are sitting randomly on M seats placed in a line. N <= M

X → occupied

0 → empty

(i-1) ← i → (i+1)

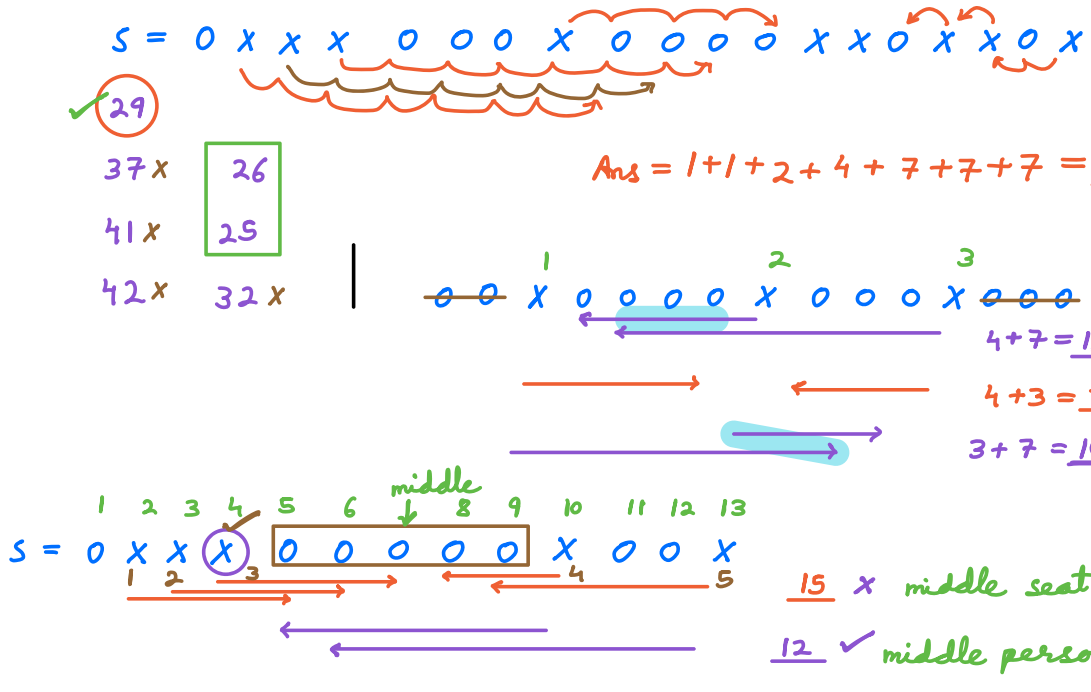
In one more a person can move to any adjacent empty seat.

Find min moves to make all people sit together.

S = X 0 0 0 X X 0 0 X 0 Ans = 5

S = X 0 0 0 X 0 0 0 X Ans = 6

S = 0 X X X 0 0 0 X 0 0 X Ans = 8


S = O X X X O O O X O O O O X X O X X O X
29
37 X
41 X
42 X
26
25
32 X
Ans = 1+1+2+4+7+7+7 = 29
1
2
3
4+7 = 11
4+3 = 7
3+7 = 10
middle
S = O X X X O O O O O X O O X
15 X middle seat
12 middle person.
To cont.