Greedy

Thursday, 30 January 2020 5:02 PM

① Given unsorted array

-2, 0, 5, -1, 2

and you are allowed $\boxed{\text{K- negations}}$
i.e. for any number
a[i] = - a[i] and
can put it back in array.
After performing k- negations, what will be the maximum sum of the array.

for k = 4

①	2	0	5	-1	2
②	2	0	5	-1	2
③	2	0	5	1	2
④	2	0	5	1	2

Maximum Sum = 10

Eg ②

	3	4	-10	2
①	3	4	10	2
②	3	4	10	-2
③	3	4	10	2

$\boxed{3 \quad 4 \quad 10 \quad 2} \rightarrow 19$

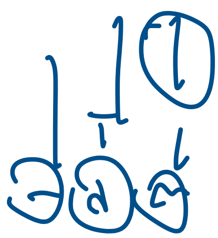
Approach 1: Sorting (Wont Work)

5 3 2 1 k = 3

Sort → 1 2 3 5

Apply negation on first three → -1 -2 -3 5

→ max Sum ✗



Observation : What if all the elements are +ve ?

→ Apply negation on minimum element

let say for -1 2 3 4
(i) 1 2 3 4

Now apply next negation on 1 only (min element)

(i) For -10 -100 0 1 2
where should the negation be applied?

We always apply negation on the minimum element.

Do we know any data structure which can give us the minimum in optimised time?

→ Min heap

→ Extract the minimum element, do a negation, put it back and then p

What if all elements are +ve?
→ We apply negation on min element.
→ After first negation, apply it again on min no.

(ii) For 0, 1, 2, 3, 4
apply negation on 0.

For +ve, -ve no. apply negation on

We can prune some steps also.

(1) If 0 is there at top of heap, no more operations are required.

(2) If no → ~~even~~ +1 & k % 2 == 0

... of more operations

(3) If $n \to -ve$ & $k \% 2 == 1$
we ~~need~~ dont need op^n

→ Same rope problem

Q/ 1 -2 -3 10 100

find subset of size K
with maximum sum

→ chose k max values of
arrays

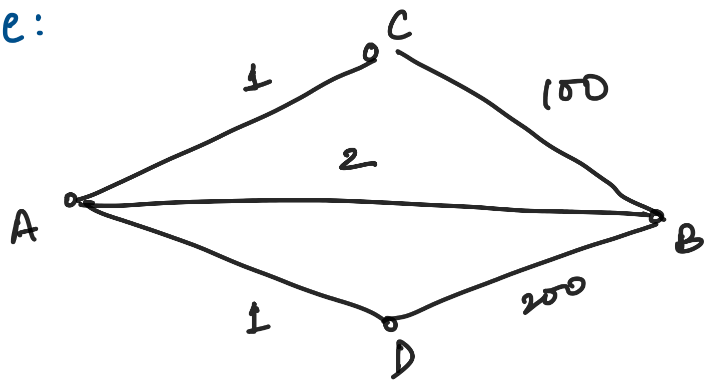
1st max = 100 $k-1$
sol^n of every step contributing to
global sol^n.

---

will greedy always work?

✱ → It only works only when local
answer contributes to global ans.

Example:



Local solution for first step → 1
Global ans → 2

+ Greedy can only work only when
local answer contributes to global
answer.

Amazon [-1 -1 -2 4 3]

Min possible subset product

24 [complete array]

-1, 0, 2 → -2 → [-1, 2]

→ [0, 1] → 0

What if all +ve no?

2 1 3 4 5

→ Ans [Min no]

→ 2 1 3 0 4 5 Min no

For all n ≥ 0 ans → min

→ What if 1 -ve no. also present?

2 1 3 0 4 5 -1

Ans = -120 -1 × $\underbrace{\quad}_{\text{all other no.}}$

→ What if 2 negative no - ?

→ choose all +ve no and -ve no. with most magnitude

2 1 3 0 4 5 -2 -1

Defining factor → no. of negative elements.

Cases

(1) ↳ No negative no → Min(A)

(2) → even -ve no's $\prod [a[i]]$

→ $a[i] \neq 0$

✗ negative no with min value

→ odd -ve elements

Take product of all no's except 0.

Greedy solution are always based on intuition and observation.

Activity Selection Problem

	a_0	a_1	a_2	a_3	a_4	a_5
st.	1	3	0	5	8	5
end	2	4	6	7	9	9

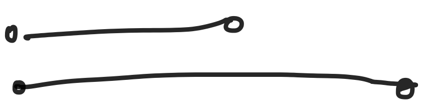
→ You can perform only activity at a time.

Find max. no of activities that can be performed.

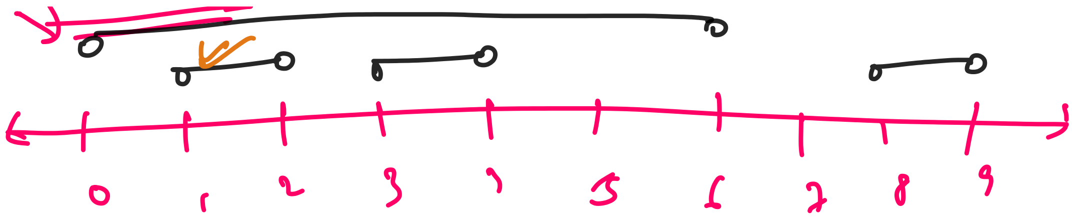
$\boxed{a_1, a_2, a_4, a_5}$ → 4

can choose activities in any order.

$\boxed{e_i < s_{i+1}}$ → constraint

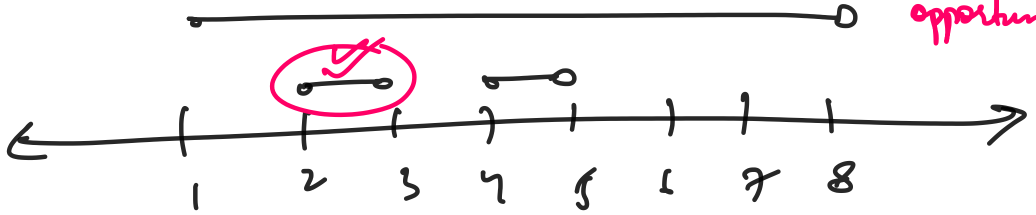


observation:



Choosing activity with ending time first,

If we choose this, we miss out on others opportunities



lets say the activities after sorting based on the ending time, we get

$s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \quad s_6$

$e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6$

$\left[\frac{s_1}{e_1}, \frac{s_3}{e_3}, \frac{s_4}{e_4} \right]$ → Prove that after choosing the activity which end first, we will get max ans.

another valid solⁿ → $\left[\frac{s_2}{e_2} \quad \frac{s_5}{e_5} \quad \frac{s_6}{e_6} \right]$

We know $e_1 < e_2$

also $e_2 < s_5$

$\boxed{e_1 < s_5}$

Time Complexity → $(n \log n)$ → Sorting

Amazon

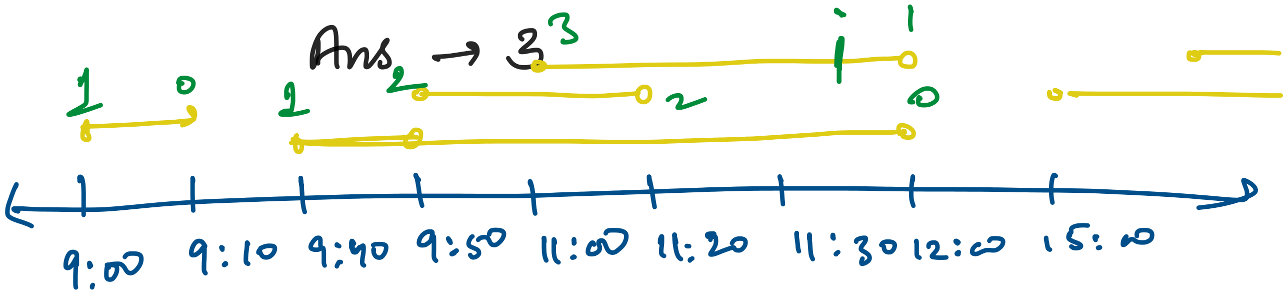
Paytm-2

Topper

A : 9:00 9:40 9:50 11:00 15:06 18:00

1) : 1:10 12:00 11:20 11:30 19:00 20:00

Find the min no. platforms required to accomodate all the trains



* Maximum no. of platforms we need at a time is the max no of trains at a time on the platform.

Matter of Concern:

If a train is leaving, I can accomodate another train after its end time.

1. Sort Arrival Time
2. Sort Departure Time

→ Keep pointers to those array and inc/dec.

Time Complexity : $O(n \log n)$

Job Scheduling

Each Job has a deadline and profit associated.

Job	Deadline	Profit
a	4	20
b	1	10
c	1	40
d	1	30

1 Job ≃ 1 day

1 Job ≃ 1 day

- Job can be performed in any order.
- Not necessary to perform all
- Maximize profit

40			20

Ans = 60

a	2	100
b	1	19
c	2	27
d	1	25
e	3	15

c	a	e

Ans → 142

✗ Sort job as per deadline ??

Aim → To do maximum no. of jobs ?? No

→ Maximize profit ?? Yes

lets say

a	1	50
b	1	100
c	1	20

Choose job with max profit

Now,

a	2	100
b	1	100
c	3	100

What should be the order??

for 3 index/ one day one, I have all the options.

→ one day 3, I have only 1 option.

If I dont perform job c on day 3, and do it any other day lets say at day 2, then I would lose out on one of the other jobs.

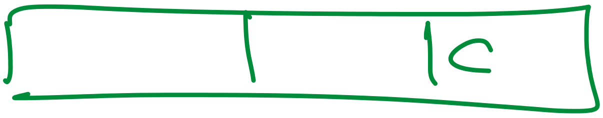
→ Always better to perform the job on its day itself.

1. Which job to choose first??
   → offering max profit

2. When to prefer it?
   As close to the deadline as possible

a	3	100
b	3	100
c	3	200

Now, losing for more profit



→ Job → 3 is gone, perform it as close as deadline possible.

---

1. sorting as per profit (dec)

② Try to find a slot

T → C

Dry Run

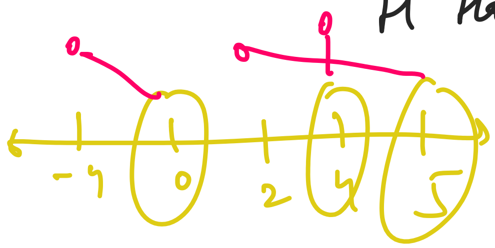
$n \simeq d$

$O(n\log n) + O(n^2)$

Closest element to x

optimising it using Balanced BST

M Mices	4	-4	2
H Holes	4	0	5



0 + 4 + 3 → 4

for

4	-4	2
4	5	0

→ 0 4 3 → 4

→ Ans depends on maximum distance that any mice has to travel.

optimal hole for mice at -4

$\mathrm{Max}(|M_1 - H_1|, |M_2 - H_2|)$

$\mathrm{Max}(|M_1 - H_2|, |M_2 - H_1|)$