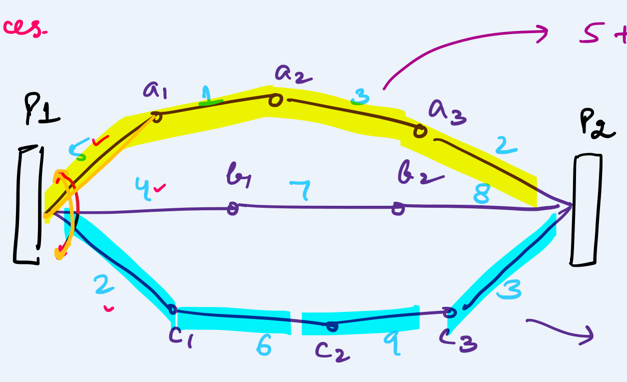
Wednesday, 15 December 2021 8:38 PM

Greedy algorithms

Greedy algorithms always make the choice that seems to be best at the moment without taking into consideration the future consequences.



$5+1+3+2 = 10$

$2+6+9+3 = 20$

final path optimised

1. Greedy algorithms are always based on intuition & observation.
2. Greedy algorithms can only work when local optimal solution contributes to global answer.

Two properties:

① Greedy choice property:

If by choosing the best choice at each step without reconsidering the previous step chosen.

② optimal substructure

If the optimal overall solution to the problem corresponds to the optimal solution of its subproblems.

Q Task scheduler (leetcode)

Total CPU cycles = $\boxed{\text{Busy}}$ + $\boxed{\text{Idle}}$?? (min)

No. of busy slots = no. of tasks execute = len(arr)

Max no. of idle slots is defined by the frequency of most frequent task.

A B A A B C A A

n = 2

A B C A B _ A _ _ A _ _ A

2 2 2 2

8 + 5 = 13

Max possible idle slots = (f_max − 1) * cooling period

= (5 − 1) * 2

= 8

A B A A B C A A

A → 5

B → 2

C → 1

8 − 2 − 1

A B C A B _ A _ _ A _ _ A

Total time = 8 + 5 = 13

A → 6

B → 6

C → 5

n = 1

A B A C A B A C A B A C B C B C B

Total → 17

Q. Minimum subset product

Eg: {−1, −1, −2, 4, 3} ans = −24

Eg: {−1, 0} ans = −1

Eg: {−1, 0, 2} ans = −2

① what if all +ve no ?? [no negative no]

[2 1 3 5]

[2 1 3 0 4 5]

ans → min ele of the array

② what if 1 −ve no. present ?? [odd no. of negative no.]

[2 1 3 0 4 5 −1]

ans → $\underbrace{-1}_{\text{negative ele}} \times \underbrace{\qquad}_{\text{all other no. except 0}}$

③ what if 2 −ve no are present ?? [even no. negative no]

[2 1 3 3 4 5 −2 −1]

ans = $\underbrace{-2}_{\text{−ve with smaller magnitude}} \times \underbrace{\qquad}_{\text{all positive except 0.}}$

Defining factor → no. of negative elements.

Cases:

1. no negative no: min (arr)
2. even no. of negative no.

$\prod_{a[i] \neq 0} [a[i]]$ / negative no. of min value

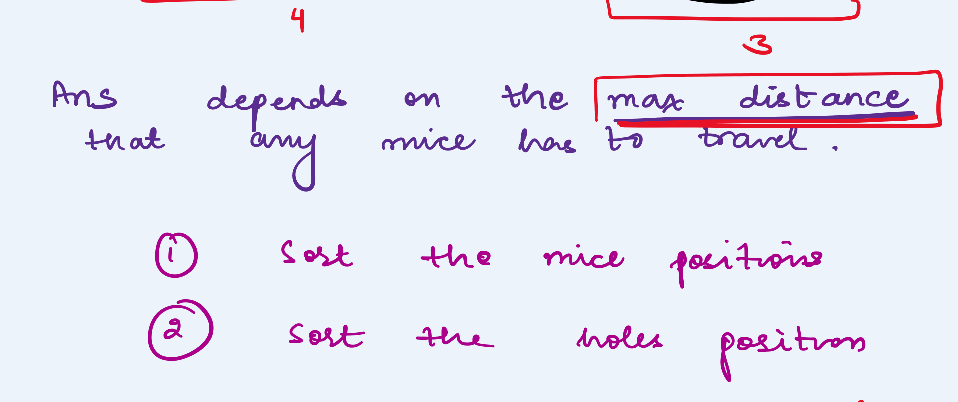
3. odd no. of negative no.

$\prod_{a[i] \neq 0} [a[i]]$

Q Mice & holes

mice = [4, −4, 2]

holes = [4, 0, 5]



Ans depends on the max distance that any mice has to travel.

① Sort the mice positions

② Sort the holes positions

| mice [i] − holes [i] |

mice → [−4, 2, 4]

holes → [0, 4, 5]

4 2 1

i = 0, |−4 − 0| = 4

i = 1, |4 − 2| = 2

i = 2, |4 − 5| = 1

max (4, 2, 1) = 4 ans

min

Q Meeting rooms II (HW)