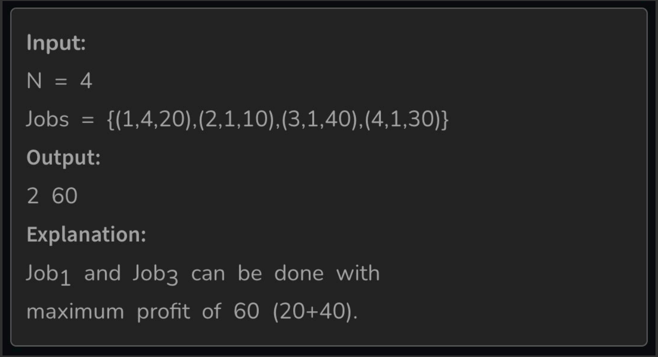
Let's start at 9:02PM

L57
Greedy Classical Problems - 2

Join Discord - https://bit.ly/ly-discord

More Classical Problems

1. Job Scheduling to maximise total profit



Input:

N = 4

Jobs = {(1,4,20),(2,1,10),(3,1,40),(4,1,30)}

Output:

2 60

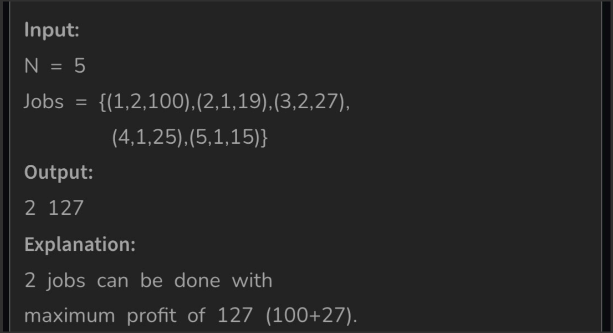
Explanation:

Job$_1$ and Job$_3$ can be done with maximum profit of 60 (20+40).

deadlines = [4, 1, 1, 1]

profits = [20, 10, 40, 30]

ans ⇒ 2 60



Input:

N = 5

Jobs = {(1,2,100),(2,1,19),(3,2,27),
(4,1,25),(5,1,15)}

Output:

2 127

Explanation:

2 jobs can be done with
maximum profit of 127 (100+27).

day 1   day 2

deadlines = [2, 1, 2, 1, 1]

profit = [100, 19, 27, 25, 15]

~~1~~, ~~2~~, 3, 4   p = 127

ans = 2 127

Intuition

1.) Highest profit first ✗

$p = [100, 90, 80]$

$d = [2, 1, 2]$

2.) Earliest deadline first, if deadline same ⇒ highest profit first ✗

$p = [10, 100, 100]$

$d = [1, 2, 2]$

3.) Prioritize doing the higher profit jobs, but only do them as late as possible.

p	~~87~~	~~100~~	~~5~~	~~21~~	~~15~~	~~26~~	~~7~~	~~10~~
d	~~3~~	~~5~~	~~5~~	~~2~~	~~4~~	~~3~~	~~6~~	~~3~~

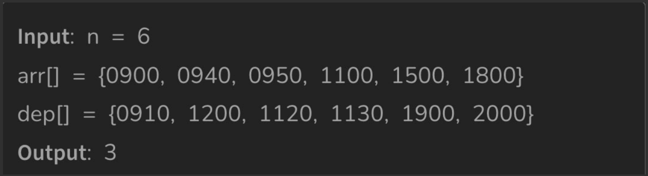
free slots ⇒ ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, 7

$p = 256$

Solution

Let’s
implement

2. Minimum Platforms



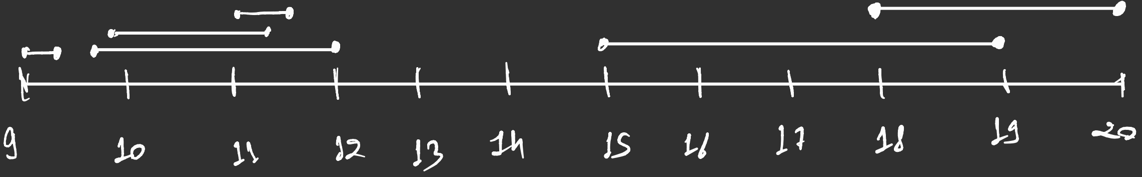
Input: n = 6

arr[] = {0900, 0940, 0950, 1100, 1500, 1800}

dep[] = {0910, 1200, 1120, 1130, 1900, 2000}

Output: 3

3 Platforms



Intuition
1

Solution 1

1.) Sort the trains by arrival time.

2.) Iterate over the trains.

3.) If smallest dep time in our DS < cur Arrival Time:

- a) Get rid of that train
- b) Reuse the platform ⇒ Add cur dep time to DS.

Else:

- a) Add the dep time of cur train to the DS
  (Create a new platform)

Let’s
implement

Intuition
2

Solution
2

Let’s
implement

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you’ll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!