Let's start at 9:05 PM

L66
Dynamic Programming : Mixed Problems 2

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

RECAP

Let’s dive right into it

1. Max Rewards - I

We've to travel for the next N days. There are N cities to travel to. On each day, we can visit only 1 city. (Note that 1 city can be visited on 2 or more days as well)

Assume that cost of travel is 0. Assume that you just teleport from 1 city to another.

For each particular day & city, there is a reward that you'll get if you go to that city on that particular.

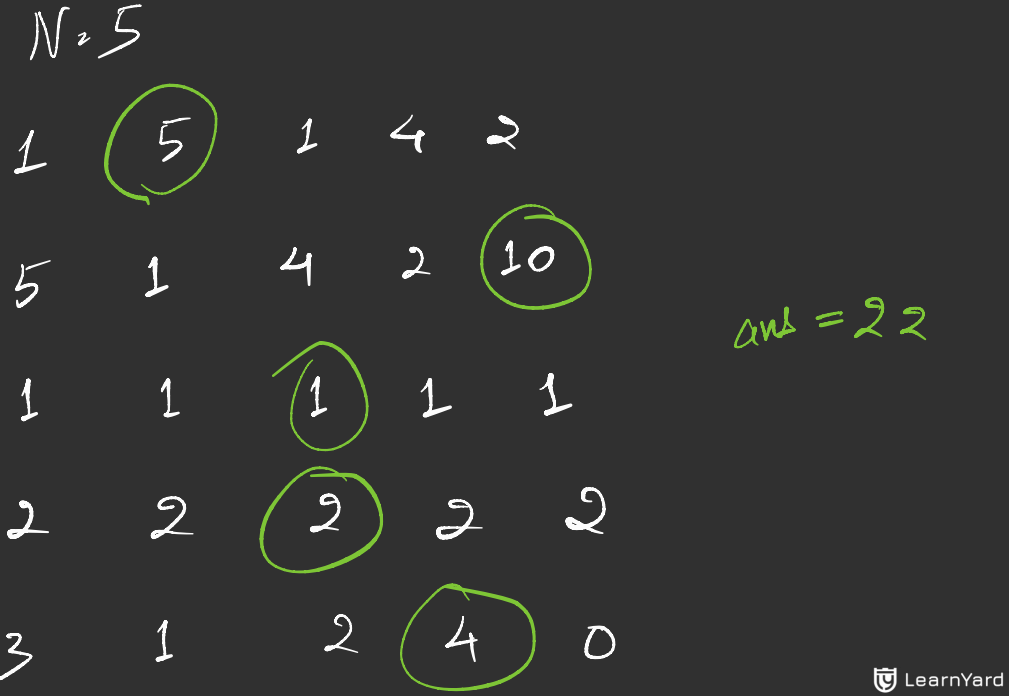
You're basically given a 2D array rew[N][N] where rew[i][j] represents the no. of reward points you'll earn if you go to j^{th} city on i^{th} day.

Find the maximum possible total reward points that we can earn.

Constraints:

$$1 \leq N \leq 10^3$$

$$1 \leq rew[i][j] \leq 10^6$$



Intuition

Be greedy!

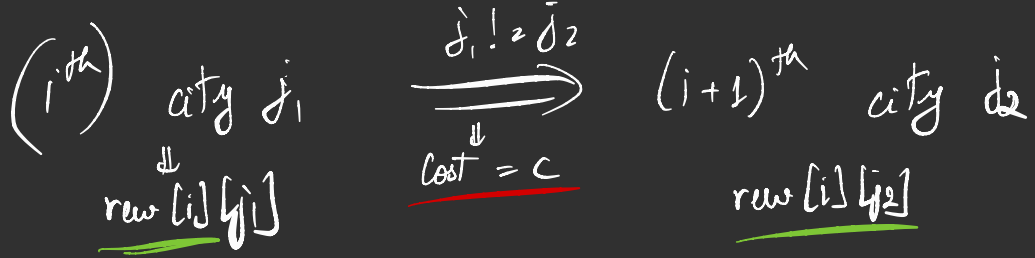
Solution

Take max from each row & keep adding.

Let’s implement

Fixed Cost C

2. Max Rewards - II



N = 5 , C = 10

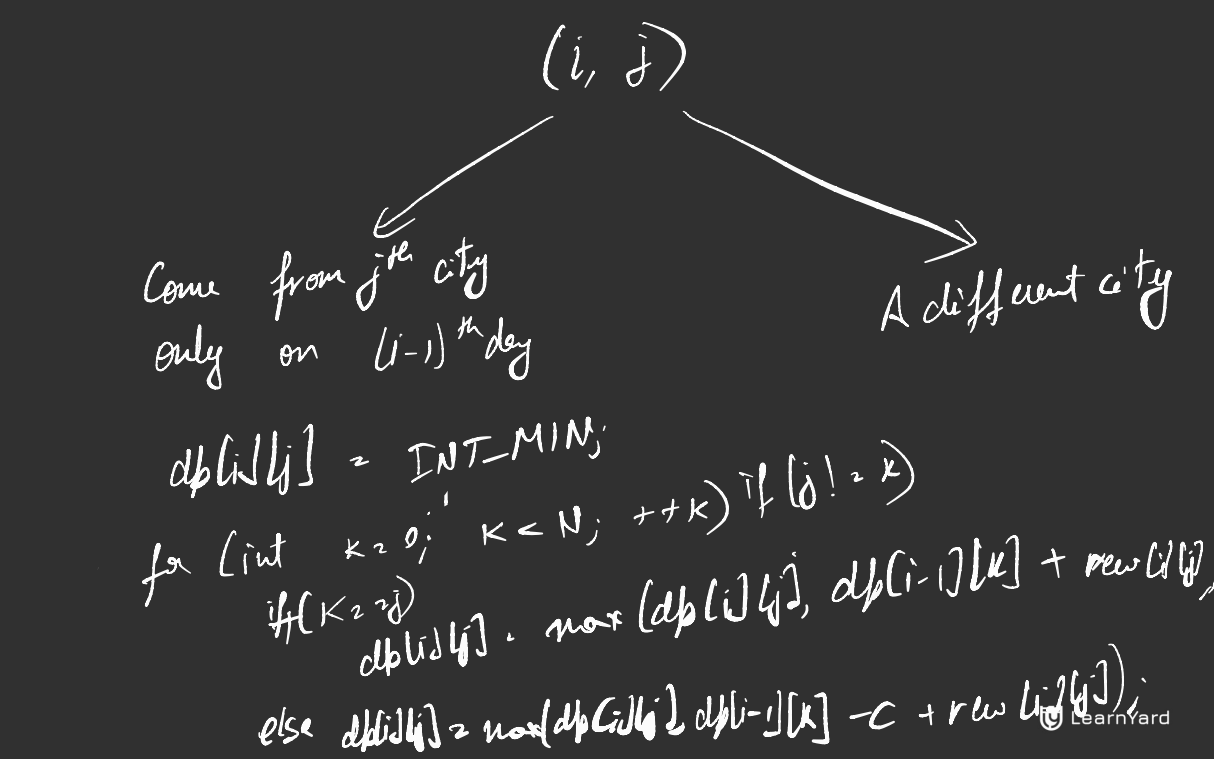
1	5	1	4	2
5	1	4	2	10
1	1	1	1	1
2	2	2	2	2
3	1	2	4	0

ans = 15

Intuition

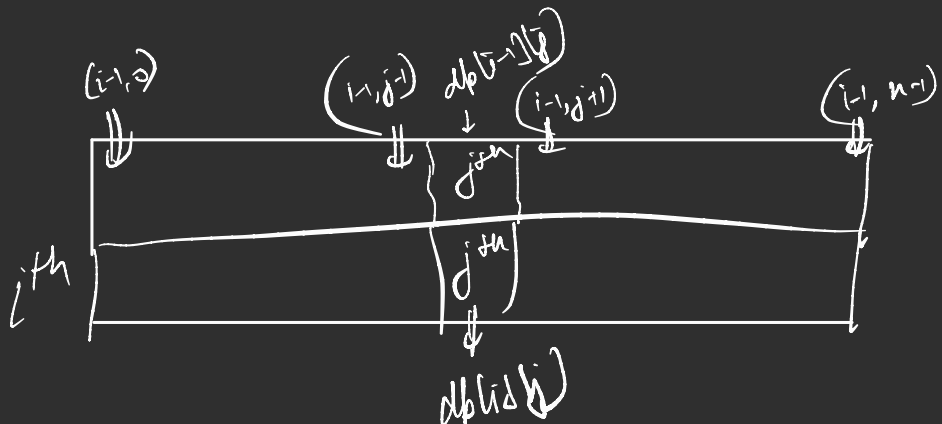
f(i, j)
(dp[i][j])

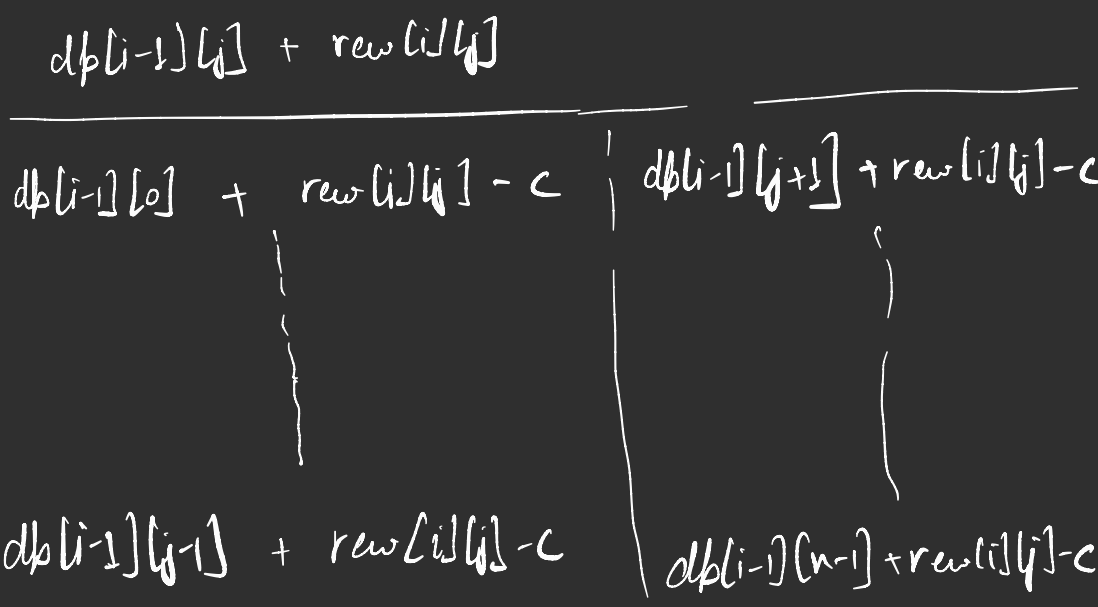
$\Longrightarrow$ Represents the max. reward that we can earn till the i^{th} day given that on i^{th} day, we travel to the j^{th} city only.

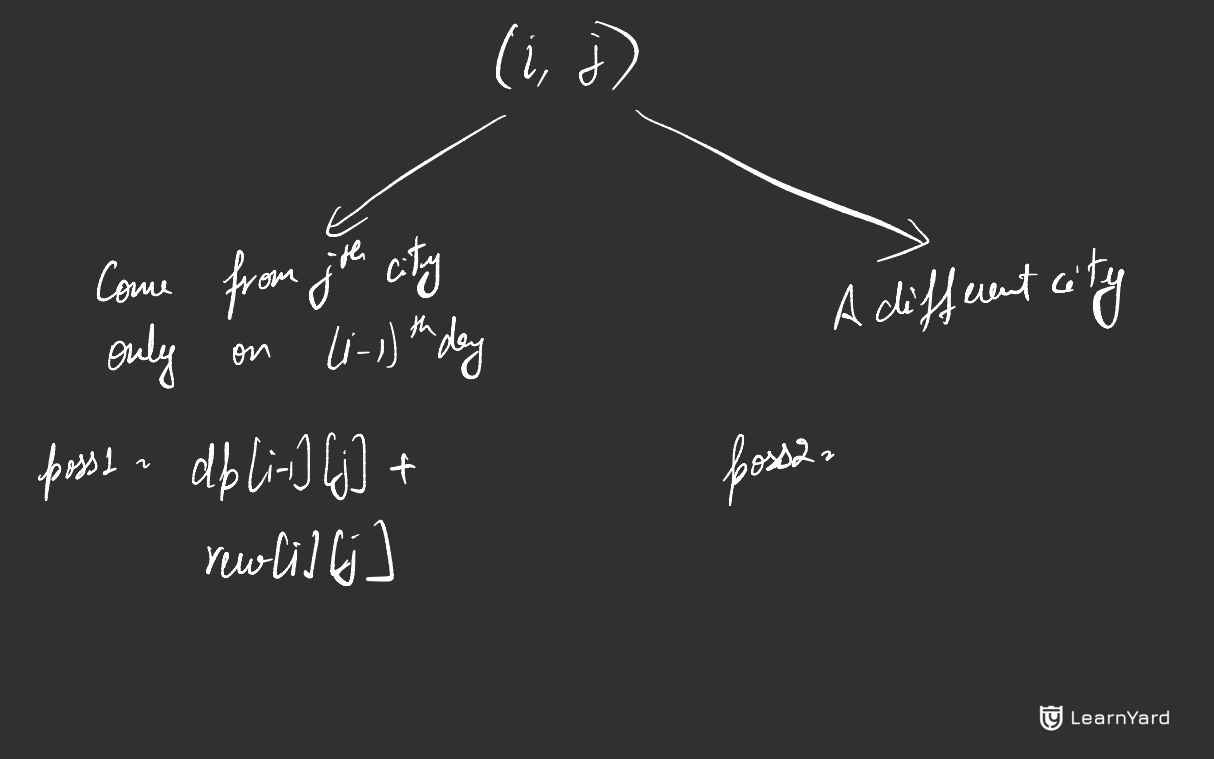

(i, j)
Come from j^{th} city only on $(i-1)^{th}$ day
A different city
dp[i][j] = INT_MIN;
for (int k = 0; k < N; ++k) if (j != k)
if(k == j)
dp[i][j] = max(dp[i][j], dp[i-1][k] + rew[i][j]);
else dp[i][j] = max(dp[i][j], dp[i-1][k] - c + rew[i][j]);

```
ans = INT_MIN;
for(j=0; j<N; j++)
    ans = max(ans, dp[N-1][j]);
return ans;
```

Overall Time $\Rightarrow$ $O(N^3)$


(i-1, 0)
(i-1, j-1)
dp[i-1][j]
(i-1, j+1)
(i-1, n-1)
jth
ith
dp[i][j]


dp[i-1][j] + rew[i][j]
dp[i-1][0] + rew[i][j] - c
dp[i-1][j-1] + rew[i][j] - c
dp[i-1][j+1] + rew[i][j] - c
dp[i-1][n-1] + rew[i][j] - c


(i, j)
Come from jth city only on (i-1)th day
A different city
poss1 = dp[i-1][j] + new[i][j]
poss2 =

Solution

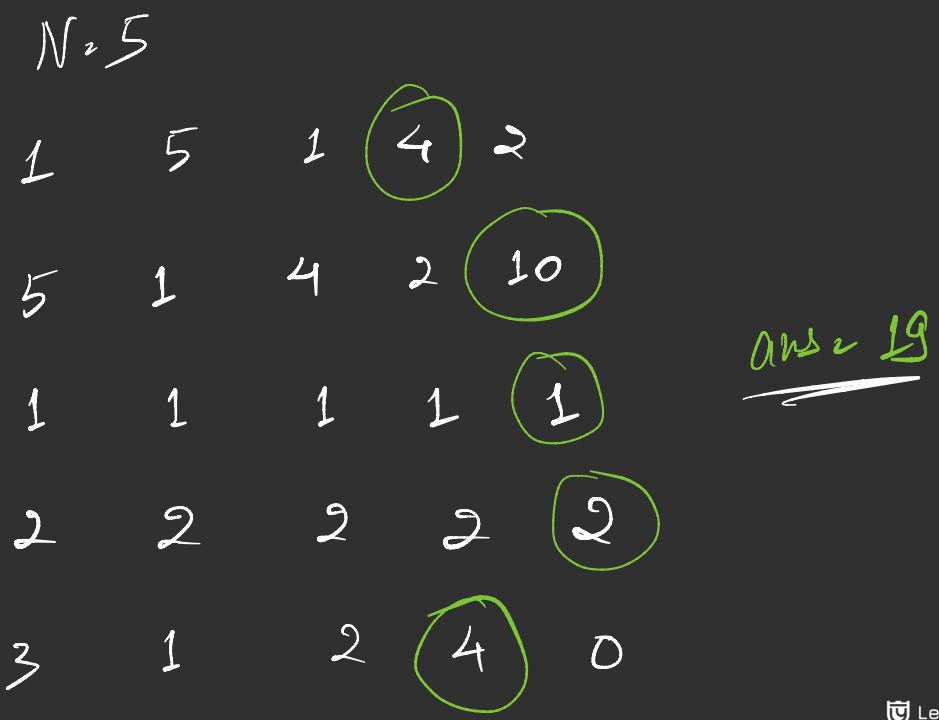
Let’s implement

3. Max Rewards - III

The cost is not fixed.

$(i, j_1) \xrightarrow{j_1 \Rightarrow j_2} (i+1, j_2)$

$\text{cost} = \text{abs}(j_1 - j_2)$



Intuition

dp[i][j] ⟹ coming from city k.

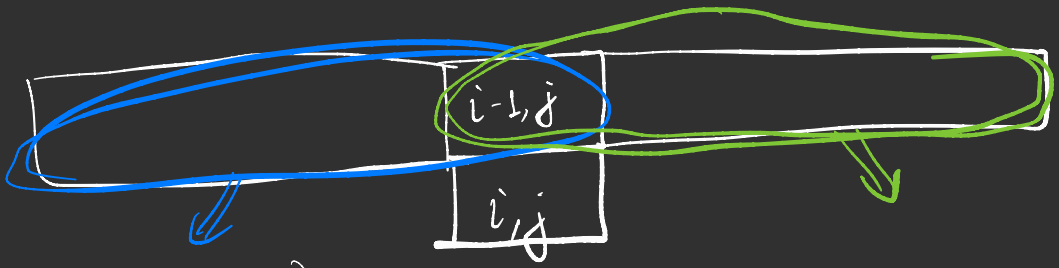
rew[i][j] + dp[i-1][k] - abs(k-j)

if (k <= j)

(rew[i][j] - j) + (dp[i-1][k] + k)

if ($k \geq j$)

$$\left(rew[i][j] + j\right) + \left(dp[i-1][k] - k\right)$$



$\max\left(\left(dp[i-1][k] + k\right)\right)$ $(k \leq j)$

$\max\left(dp[i-1][k] - k\right)$ $k \geq j$

poss1 = rev[i][j] - j + pre[i-1][j]

poss2 = rev[i][j] + j + suf[i-1][j]

dp[i][j] = max(poss1, poss2);

Solution

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!