Tutorial-9

● Graded

Student

Abhinav Shripad

Total Points

2.75 / 3 pts

Question 1

(no title) 2.75 / 3 pts

- ✔ **+ 0.75 pts** Identifying that there are constant ways of placing pebbles in a column - 0.75 points
- ✔ **+ 1 pt** Correct recurrence relation - 1 point
- ✔ **+ 0.25 pts** Correct Base Case - 0.25 points
- ✔ **+ 0.25 pts** Brief justification of the recurrence relation - 0.25 points
- ✔ **+ 0.25 pts** Mentioning the correct order of filling the DP table - 0.25 points
- **+ 0.25 pts** Outputting the optimal placement (not just the optimal value) - 0.25 points
- ✔ **+ 0.25 pts** Brief justification of the time complexity - 0.25 points
- **+ 0.6 pts** Written "I do not know how to approach this problem" - 0.6 points
- **+ 0 pts** Incorrrect

COL351: Analysis and Design of Algorithms
Tutorial 9

Name: Abhinav Rajesh
Entry number: 2022CS11596

Date: Oct 03, 2024
Group: 3

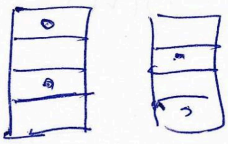
Observe that each ~~colaumn~~ column can have at max 2 pebbles - If not, then having 3 or 4 pebbles, 2 will be adjacent.

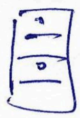
For each column there are

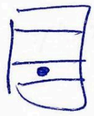
$\binom{4}{1} + \binom{4}{2} = 4 + 6 = 10$ possible configurations

let them be $w_1, w_2 \cdots w_{10}$.

for w_i, w_j define compatible(w_i, w_j) = true if w_i and w_j can be neighbours.

ie  ⇒ compatible

and let index(w_i) be the indices of row cell in w_i. eg index() = 1, 3

index() = 3, index() = 4

$dp(i, w_j)$ = maximum value if only column $[1:i]$ are considered and column i is in state w_j

$$dp(1, w_i) = \sum_{x \in index(w_i)} A[1, x] \quad \# \text{ Base case}$$

$\cancel{dp(i, w_j) = \max_{\text{compat}}}$

$\cancel{dp(i, w_j) = \max_{\text{compatible}(w_k, w_j)}}$ (A ~~[illegible]~~

$$dp(i, w_j) = \sum_{x \in \text{index}(w_j)} A[i, x] + \max_{\text{compatible}(w_k, w_j)} (dp(i-1, w_k))$$

$i > 1$

$$\text{answer} = \max_{1 \le i \le 10} (dp(n, w_i))$$

$$T \cdot C = O(n|W|^2) = O(n) \quad , \quad S \cdot C \cdot = O(n|W|) = O(n)$$

$\underline{|W|} = 100$, $\underline{|W|} = 10$

<u>Proof of correctness</u>:

if last column is in state w_j, then $(n-1)^{th}$ column can take state w_k such that w_k and w_j are compatible and answer till $n-1^{th}$ colomn ~~oos~~, with last state w_k is max.

Thus justifying the recurrence.