Q-6.2

Given: Initial mile post $a_0 = 0$, destination mile post a_n; and n mile posts: $a_1 < a_2 < \ldots < a_n$ where we could stay.
Ideal travelling distance is 200 miles a day.
Penality $(200 - x)^2$ where x mile is distance travelled that day.

Let's say
$mtP(a_i)$: returns minimum total Penality to reach a_i mile post hotel; $mtP(a_0) = 0$
we need to find $mtP(a_n)$; lets say need to stay at hotel at before reaching a_n.
define mile array $A[n-1]$.
for $i = 1$ to $i = n-1$.

$$A[i-1] = mtP(a_{i-1}) + (200 - (a_n - a_i))^2$$

Now $mtP(a_n) = min(A[0, 1, \ldots n-1])$

If mile post sequence is same we could use Array A if further a_{n+1} is added, Hence dynamic of a recurrence. we need array to store A, a_i, $mtP(a_i)$ each of size n.

---

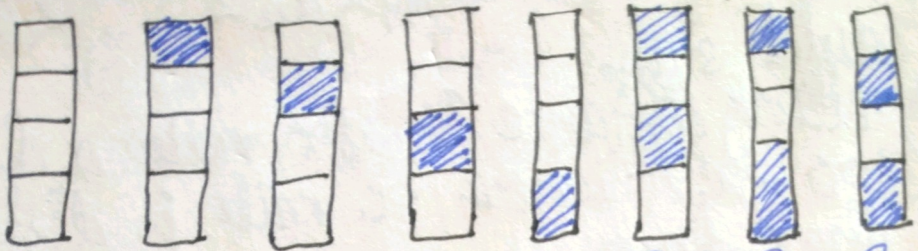
for each i we need almost $O(n-1)$ time Hence overall complexity

$$\sum_{i=1}^{n} O(n) = O(n^2)$$

$$\sum_{i=1}^{n} O(n-1) = nO(n-1) = O(n^2 - n) = O(n^2)$$

Q-6.5 n columns, 4 rows, 20 pebbles.
constraint: No vertical or horizontal adjacency.
also maximize total sum.

(a)



These are 8 legal Patterns as shown.

If P_i and P_j could be placed adjacently then pair is compatible with and each other.

(b) $lpS(P_i, K) =$ maximum Sum of Integers through legal pebbling till K^{th} column, where last $(K-1)^{th}$ column is P_i

$Sum(P_i)$ = Sum of integers pebbled in pattern P_i.

for $K = 1$ to n

$$lpS(P_i, K) = Sum(P_i) + \max_{j} lpS(P_j, K+1)$$

where P_i, P_j are compatible

Time Complexity $O(n) \cdot O(8) = O(n)$

We need arrays to store $4 \times n$ Integers, P_8 Patterns, n size lpS.. etc.

Base case $lpS(P_i, 0) = 0$, it will be added to find further for $n \geq 0$ problems.

Q6.10 | $O(n^2)$ | given $n \rightarrow$ distinct coins (biased) with probability $P_1, P_2 \ldots P_i \ldots P_n \in [0,1]$ where P_i is the probability of getting Head over i^{th} coin (say x_i) here lets $x_1, x_2 \ldots, x_n$ are coins

$\Rightarrow Pr(x_i = 1) = P_i$; $Pr(x_i = 0) = 1 - P_i$

we need to find

$Pr(x_1 + x_2 + \ldots + x_n = K)$ = Probability of exactly K heads using (tossing) n-coins $(x_1 \ldots x_n)$

Dividing to subproblems.

$$Pr\left(\sum_{i=1}^{n} x_i = K\right) = Pr\left(\sum_{i=1}^{n-1} x_i = K\right) \cdot (1 - P_n) +$$

$$Pr\left(\sum_{i=1}^{n-1} x_i = K-1\right) \cdot P_n$$

$\Rightarrow$ when K^{th} Head occurs before x_n OR

K^{th} Head occured at nth coin (x_n).

PR[n+1][K+1] 2-D array; elements PR[i][j] returns Probability $Pr\left(\sum_{r=1}^{r=i} x_i = j\right)$.

PR[0][0] = 0

for i = 1 to n :

BaseCases: For j = 1 to K :

BaseCases : If (i < j // i = j = 0) Break;

PR[i][j] = PR[i-1][j] (1 - P_i) +

PR[i-1][j-1] P_i

return PR[n][K].

Time Complexity $O(n^2)$, BaseCase PR[0][0] = 0

Q6.10 | $n \log^2 n$ given n – distinct biased coins, $P_i \in P_1 \dots P_n \in [0,1]$, k, as explained above.

$$f(x) = \prod_{i=1}^{n} (1 - P_i + P_i x) = (1 - P_1 + P_1 x)(1 - P_2 + P_2 x) \dots (1 - P_n + P_n x)$$

we need to find coefficient of x^k of polynomial $f(x)$.

Q6.14 cloth dimention x, y; $x, y \in I^+$

n-products dimention $a_i \times b_i$ and cost c_i for $i \in [1, n]$ where $a_i, b_i, c_i \in I^+$ (positive integer)

⇒ we need to find maximum sum of selling price.

TSP(x,y) :- returns maximum total selling price using cloth of dimention x, y with constraints.

Booleanfn(x, y, a[ ], b[ ]) :- returns True(1) or false(0)

for $i \in [1, n]$, B = false(0)

if x == a_i and y == b_i ~~return true (1)~~

B = True(1) Break;

else B = False(0);

Return B

---

costfn(x, y, a[ ], b[ ], c[ ]) : returns cost if no further pieces needs to be made for x, y dimention cloths.

cost = 0

for $i \in [1, n]$

if x == a_i and y == b_i

cost = c_i, Break | else cost = cost.

Return cost.

---

making partitions

TSP(x, y)

def maxA = 0, A[0, 1, 2, ... x], A[0] = 0

For x = 1 to x = x

A[x] = TSP(x, y) + TSP(x − x, y)

if (maxA ≤ A[x]) then maxA = A[x], A[0] = x.

def maxB = 0, B[0, 1, 2, ... y], B[0] = 0

For y = 1 to y = y

B[y] = TSP(x, y) + TSP(x, y − y)

if maxB ≤ B[y] then maxB = B[y], B[0] = y.