Evan Zhang | z5383657

Question 1

1.1

<u>Subproblems:</u>

For each row $1 \leq i \leq m$ and column $1 \leq j \leq n$, let P(i,j) be the problem of determining opt(i,j), the maximum total coins in a path up to and ending on the square (i,j).

<u>Recurrence:</u>

opt(i,j) = max(opt(i, j - 1), opt(i - 1, j)) + $v_{i,j}$, where $v_{i,j}$ is the number of coins contained in the square (i,j). There are two paths to get to square (i,j), coming from the left or coming from above. We choose the square with the greatest number of coins. Therefore the maximum number of coins to square (i,j) will be the maximum of the square to the left and the square above plus the coins contained in square (i,j).

If i = 1, opt(i,j) = opt(i, j - 1) + $v_{i,j}$. Since you can only get to this square by moving right, the maximum number of coins to the current square will be given by the maximum number of coins to the square to the left plus the coins contained in the current square.

If j = 1, opt(i,j) = opt(i - 1, j) + $v_{i,j}$. Similar to the above, where you can only get to this square by moving down. The maximum number of coins to get to this square will therefore be the maximum to get to the square above plus the coins contained in the current square.

opt(i, j - 1) and opt(i - 1, j) can be solved by similar logic to solving for opt(i,j).

<u>Base case:</u>

If i = 1 and j = 1, opt(i,j) = 0. There are no coins in the starting square.

If i = m and j = n, $v_{i,j}$ = 0. There are no coins in the finishing square.

<u>Order of computation:</u>

Subproblem P(i,j) depends only on earlier subproblems (P(k,l), where k < i and l < j), so we can solve the subproblems in increasing order of i and j. We will solve the grid of squares diagonally from bottom left to top right.

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8
5	6	7	8	9

<u>Final answer:</u>

The maximum total coins is given by opt(m,n).

Time complexity:

There are O(mn) problems/subproblems, each solved in O(1). Thus, the overall time complexity is O(mn).

1.2

Subproblems:

For each row $1 \le i \le m$ and column $1 \le j \le n$, let P(i,j) be the problem of determining p(i,j) and E(i,j), the probability of being in square (i,j) and the expected value of the total coins accumulated when in square (i,j).

Recurrence:

p(i,j) = p(i, j - 1)*p + p(i - 1, j)*(1 - p). Since you can only go into square (i,j) from either the square to its left or the square above, the probability of being in square (i,j) is given by the probability of being in the square to the left and going right or being in the square above and going down.

If i = m, p(i,j) = p(i - 1, j). Similar to the above, where if the only one possible square to move to is (i,j), the probability of going to the square (i,j) is the probability of being in the square previous times the probability of moving to square (i,j) (i.e. p(i - 1, j) * 1).

If j = n, p(i,j) = p(i, j - 1). If the only one possible square to move to is (i,j), the probability of going to the square (i,j) is the probability of being in the square previously times the probability of moving to square (i,j) (i.e. p(i, j - 1) * 1).

p(i, j - 1) and p(i - 1, j) can be solved by similar logic to solving for p(i,j).

E(i,j) = E(i, j - 1) + E(i - 1, j) + $v_{i,j}$*p(i,j), where $v_{i,j}$ is the coins contained in square (i,j). Since you can only go into square (i,j) from either the square to its left or the square above, the expected value of total coins will be the expected value of both streams plus the coins contained inside (i,j) times the probability of being in square (i,j).

If i = 1, E(i,j) = E(i, j - 1) + $v_{i,j}$*p(i,j) . If i = 1, square (i,j) is only accessible by one stream/square.

If j = 1, E(i,j) = E(i - 1, j) + $v_{i,j}$*p(i,j). Similar to the above, if j = 1, square (i,j) is only accessible by one other stream/square.

Base case:

If i = 1 and j = 1, p(i,j) = 1 and E(i,j) = 0.

If i = m and j = n, $v_{i,j}$ = 0.

Order of computation:

Same as 1.1. Subproblem P(i,j) depends only on earlier subproblems (P(k,l), where k < i and l < j), so we can solve the subproblems in increasing order of i and j. We will solve the grid of squares diagonally from bottom left to top right.

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8
5	6	7	8	9

<u>Final answer:</u>

The expected number of coins is given by E(m,n).

<u>Time complexity:</u>

There are O(mn) problems/subproblems, each solved in O(1). Thus, the overall time complexity is O(mn).