CS301 Assignment 4

Nofil Iqbal 28631

(a) **Recursive formulation** of this agricultural robotics problem.

The algorithm work like this: we will first create a matrix with the same dimensions to store the values given in the algorithm. When the robot arrives at a block, it will assign a value to for that block. This value will be generated by taking the maximum of the values to the left and top blocks of the current position of the robot, and adding 1 if there is a weed at the current block and 0 if there is no weed at the current block. Recursively doing this will give us a matrix with values of the maximum number of weeds we can remove while reaching that block. The recursive formula for this will be as follows:

$m(i,j) = \max(m[i-1,j], m[i,j-1]) + w(i.j)$

(b) **Pseudocode of your algorithm** designed using dynamic programming based on the recursive formulation.

Given matrix w[n,m]

```
a = len(w)
b = len(w[1])
m = np.zeros((a,b))
Result = [];
for i in range(0,a):
    for j in range(0,b):
        m[i][j] = max(m[i-1][j] , m[i][j-1]) + w[i][j];
weeds = m[a-1][b-1]
x = 0;
y = 0;
```

```
Result.append((x,y))
while (x != a-1 or y != b-1):
    if (x != a-1 and y = = b-1):
        y = y + 1
    elif (x == a-1 and y != b-1):
        y = y + 1
    elif (x != a-1 and y != b-1):
        if (m[x + 1][y] > m[x][y+1]):
        x = x + 1
  elif (m[x+1][y]<m[x][y+1]):
        y = y + 1
  else:
        x = x + 1
    Result.append((x,y))
```

(c) **Asymptotic time and space complexity analysis** of your algorithm.

Time Complexity = O(n*m)

Space Complexity = O(n*m) + O(n+m)