CS 301 - Assignment 4 (Report)

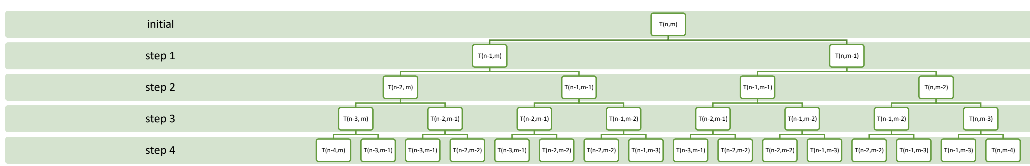
Emir Ay

December 2022

1 Recursive formulation

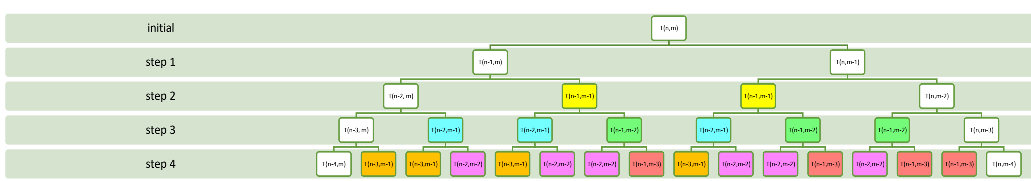
The agricultural mobile robot that can navigate one block right or one block down on a rectangular farm divided into nxm square blocks. Since the robot can only go to right or down, it will have 2 choice at each recursion, hence its recursive formulation will be $T(n) = 2T(n-1) + \Theta(1)$. Its recursion tree will be like the figure below:

Recursion tree of agricultural robotics problem



However, we can observe overlapping subproblems in this recursion tree. These overlapping subproblems might be seen clearer in the figure below:

Colored recursion tree of agricultural robotics problem (colors indicate overlapping subproblems)



In this figure, same colors indicates the same subproblems. To prevent the overlapping subproblems we need to use memoization method which is possible with a dynamic programming algorithm. As a result, we can reach the recursive formula: $T(n,m) = max(T(n-1,m), T(n,m-1)) + \Theta(1)$.

2 Pseudo-code of the algorithm

The pseudocode using dynamic programming based on the recursive formulation is below:

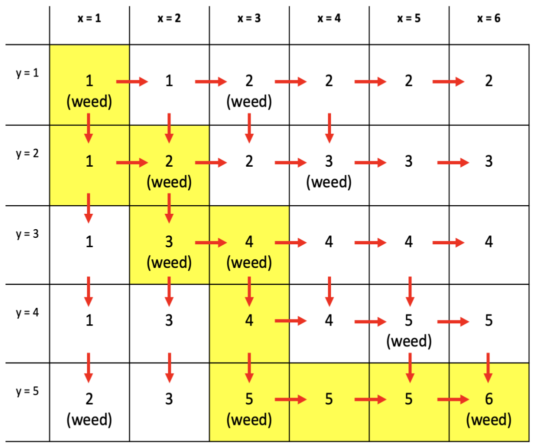
```
A: stands for the matrix for memorization

AMR(i, j)
     if there is a weed:
          A[i][j] = 1+max{AMR(i-1,j), AMR(i,j-1)}
     else:
          A[i][j] = max{AMR(i-1,j), AMR(i,j-1)}
```

Firstly, dynamic programming algorithms generally are useful for the problems have overlapping subproblems. As I showed in the first part, the Agricultural Mobile Robotics problem has overlapping subproblems and to reduce its time complexity we need to use memoization technique. To indicate it clearer, I have attached the illustration of the dynamic programming table for this problem (for the sample case in the homework document):

	x = 1	x = 2	x = 3	x = 4	x = 5	x = 6
y = 1	1 (weed)	1	2 (weed)	2	2	2
y = 2	1	2 (weed)	2	3 (weed)	3	3
y = 3	1	3 (weed)	4 (weed)	4	4	4
y = 4	1	3	4	4	5 (weed)	5
y = 5	2 (weed)	3	5 (weed)	5	5	6 (weed)

In this case, memoization is used to show alternative paths for agricultural mobile robot which can navigate one block right or one block down on a rectangular farm divided into nxm square blocks. As can be seen in the table there are paths other than the optimum solution; however, with the help of memoization it will always choose optimum path. Another illustration that shows the optimum solution is attached below:



3 Asymptotic time and space complexity analysis

For the asymptotic time complexity, I have shown that this problem has overlapping subproblems and memoization helps us to reduce time complexity by preventing overlapping subproblems. For my dynamic programming algorithm, problem states that the agricultural mobile robot that can navigate one block right or one block down on a rectangular farm divided into nxm square blocks. Therefore, asymptotic time complexity of my algorithm is O(nxm). As can be observed from the recursion tree in part 1, time comple without memoization is O(2^n). Hence with the help of memoization, my dynamic programming algorithm reduces time complexity from exponential time to O(nxm).

When it comes to space complexity, I store the data in a matrix of size nxm for memoization; therefore, the space complexity is also O(nxm).