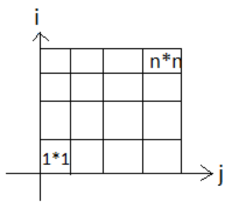
Homework Assignment 5

CS 430 Introduction to Algorithms

Name : Karthik Mahesh

CWID : A20383027

1.Solution:



At every city, Driver have 2 options ie, either to go up or to go right

As the driver can go right or up from a given city, for city (i,j) we can move to (i+1,j) and (i,j+1).

Del[i][j] = deliveries in city (i,j)

MaxDel[i][j] = maximum deliveries done by reaching (n,n) from (1,1).

MaxDel(i,j) = Del[1][1] if i&j = 1 ie, if only 1 city

MaxDel(i, j) = Del[1][j] if i=1 ie, first row

MaxDel(i, j) = Del[i][1] if j=1 ie, first column

Recursion :
MaxDel(i, j) = max(MaxDel(i-1, j), MaxDel(i, j-1)) + Del[i][j] if i & j > 1

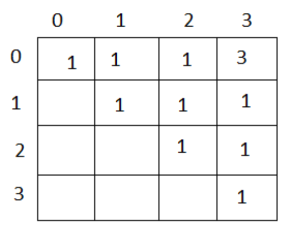
We can reach (i,j) from (i-1,j) or (i,j-1), thus, the total Deliveries by reaching (i,j) is the sum of maximum of the two deliveries plus the total deliveries inside the city (i,j)

Time complexity is **O(n^2)**

2.Solution:

We can create a matrix of the size (string length * string length) and start filling the cells by checking if the letters are same or different.
For example is the string is 'abca', we will create a matrix of size 4 x 4.

0	1	2	3
a	b	c	a



	0	1	2	3
0	1	1	1	3
1		1	1	1
2			1	1
3				1

lets start filling diagonally. The diagonal line from (0,0) to (3,3) will always be 1 as its only 1 letter.
Then we start checking with the next letters if the letters are same. If its same, we will add 2 to the palindrome inside those 2 letters.
When checking ‘a’ with ‘a’, we know it’s the same and we will see if there is any palindrome inside by checking the matrix. As there is no other palindrome and all the values are only 1, we will add 1 to the 2 and fill as 3.

After filling all the cells, we will get to know that the maximum sized subsequence that is a palindrome is 3.

```
Algo mssp(str)

      for i in range(strlen)
          value[i][j] = 1

      for sub_strlen in range(2, strlen+1)
         for i in range(0, strlen – sub_strlen + 1)
            j = i + sub_strlen - 1
            if string[i] == string[j] and sub_strlen == 2
               value[i][j] = 2
            elseif string[i] == string[j]:
               value[i][j] = 2 + value[i+1][j-1]
            else
               value[i][j] = max(value[i][j-1], value[i+1][j])
```

Time complexity will be **O(n^2)**

References used :
Analysis and design of algorithms textbook