Row and Column Wise sorted subsequences

Group 29 :
Harsh Kedia - IIB2019028
Milap Anwani - IIB2019029
Kandagatla Meghana Santhoshi - IIB2019030

Department of Information Technology
Indian Institute of Information Technology, Allahabad

Content

Abstract

In this paper, we have devised an algorithm which when given a random matrix , it gives Row Wise sorted subsequences
And Column Wise sorted subsequences.

Introduction

This document describes the procedure followed to find the Row Wise sorted subsequences for every row and Column Wise sorted subsequences for every column.

.

Algorithm description and analysis

Driver function:

1) we are generating a random matrix in the main and is sent to driver function as input.

2) In driver function we are initializing a vector of int "ls1" as each row one by one is forwarded to find function to calculate the sorted sub sequences.

3)For each row after computation it is stored in 2D vector 'st'.

4)To find column wise, we find store each column as rows in trans matrix and then similarly we compute sorted subsequences for each column in matrix or each row in trans matrix.

5)In this way we compute sorted subsequences for a matrix , row-wise and column-wise.

Algorithm description and analysis

Find function :

1)In find function, if input row is empty and output row is not empty then, storing the result in 2D vector "st" and return.

2) Else , initialize temp vector and push back value first element of row and erase that value from input row.

3) Call find function and if output row is empty then call it again with output row as temp.

4)Else if first element of temp is greater then last element of output row , then push back that temp value in output row.

5) Again Call find function

Pseudo Code

Driver Function:

For : i − > 0 to R

initialise vector<int>ls1

ls1=matrix [i]

Declare vector<int>ls2

find (ls1 , ls2)

print st

clear st

For : i−>0 to C

For : j −> 0 to R

transmatrix [i][j] = matrix [j][i];

For : i − > 0 to C

initialise vector<int>ls1

ls1=transmatrix [i]

Declare vector<int>ls2

find (ls1 , ls2)

print st

clear st

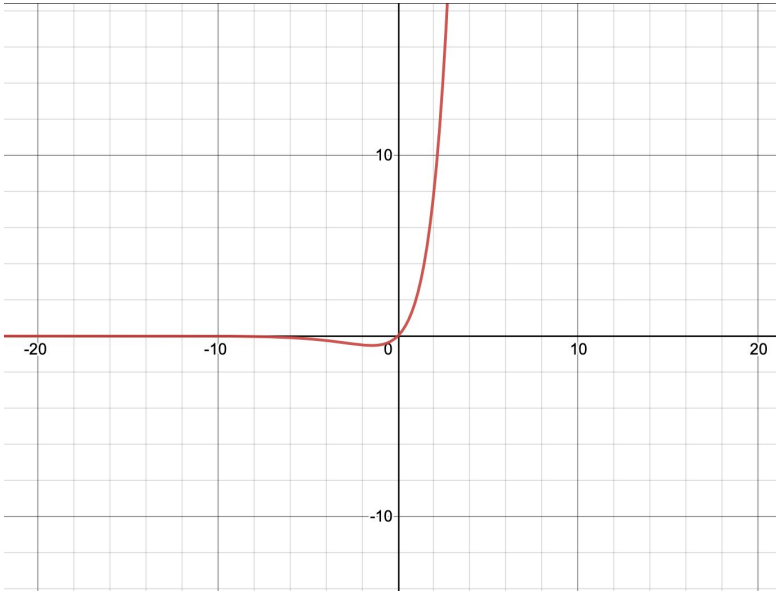
Pseudo code

Find function:

```
if ( inp . empty()&&out . size ()!=0)

st . push back ( out )

 else return

intialise vector temp

 temp . push back ( inp [0])

 erase the first element in inp

 find (inp , out )
if ( out . size () == 0)
find (inp , temp)
else (temp[0] > out [ out . size () − 1])
out . push back (temp [0])
find (inp , out )
```

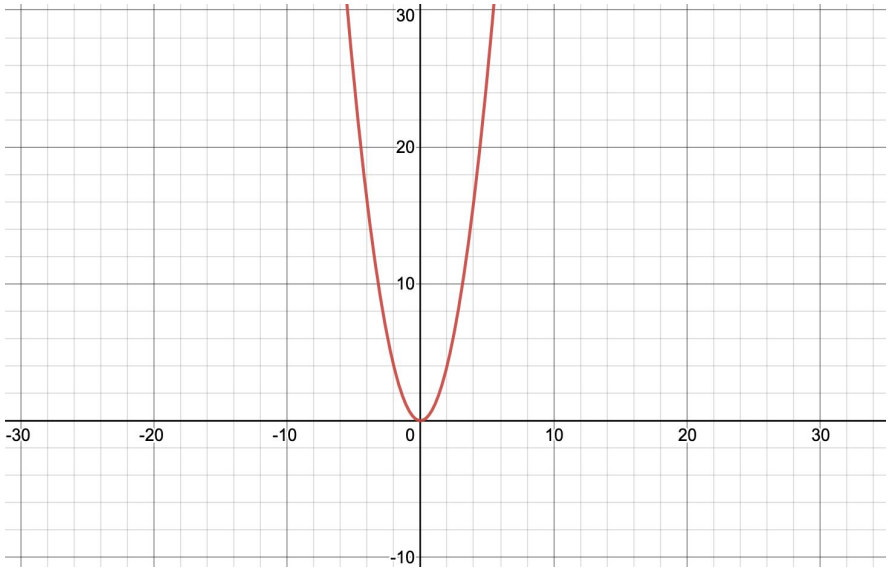
Time Complexity

The time complexity of this algorithm is O(n * 2^n).Where n is Maximum of Row and column of the randomly generated matrix.



Space Complexity

The space complexity of the Program is O(n^2).Where n is Maximum of Row and column of the randomly generated matrix.



Conclusion

We have arrived at the solution of the problem by recursion, where we are operating on each row and column, thus the time complexity is $O(n \cdot 2n)$.

References

[1] https://www.codespeedy.com/generate-a-matrix-of-random-numbers-incpp/

[2] https://www.geeksforgeeks.org/number-of-longest-increasingsubsequences

[3] https://www.inf.unibz.it/calvanese/teaching/06-07-ip/lecturenotes/uni11.pdf

THANK YOU