# National University of Computer and Emerging Sciences, Lahore Campus

| | | | |
|---|---|---|---|
| Course: | Design & Analysis of Algorithms | Course Code: | CS-2009 |
| Program: | BS (Computer/Data Science) | Semester: | Fall 2022 |
| Duration: | 60 Minutes | Total Marks: | 25 |
| Paper Date: | 10-Nov-22 | Section: | ALL |
| Exam: | Midterm 2 | Page(s): | 6 |
| Name | | Roll Number | |

**Instruction/Notes:** Ample space is provided for rough work; NO EXTRA sheets will be provided.

| Question | 1 | 2 | 3 | Total |
|---|---|---|---|---|
| Marks | /8 | /10 | /7 | /25 |

**Q1)** Consider the following recursive algorithm. [2 + 5+ 1 = 8 Marks]

SOLUTION

```
/* m and n are lengths of char arrays X and Y respectively */
int Function( char *X, char *Y, int m, int n )
{
  if (m == 0 || n == 0)
    return 0;
  if (X[m-1] == Y[n-1])
    return 1 + Function (X, Y, m-1, n-1);
  else
    return max (Function (X, Y, m, n-1) +1,  Function (X, Y, m-1, n) +1);
}
```

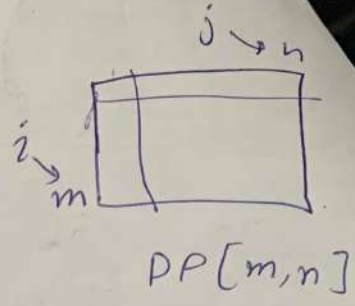
(a) What is time complexity of above algorithm? Show all working.

$O(2^{n+m})$

(b) Convert the recursive code given above into bottom up iterative dynamic progr

```
DP[0..m, 0..n]
FOR(i=0 to m)
    DP[i,0] = 0
FOR( j= 0 to n)
    DP[0,j] = 0

FOR ( i = 1 to m)
    FOR ( j = 1 to n)
        IF (X[i-1] == Y[j-1])
            DP[i,j] = 1 + DP[i-1, j-1]
        ELSE
            DP[i,j] = max { 1 + DP[i, j-1]
                          { 1 + DP[i-1, j]
```



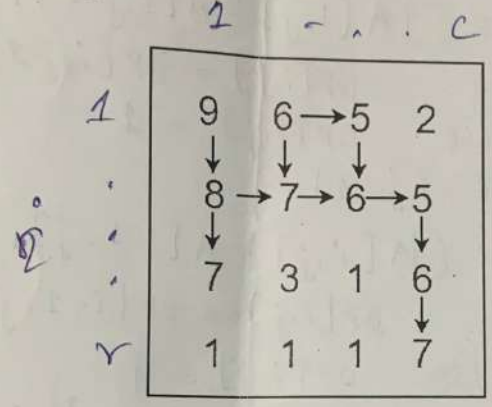

(c) What is time complexity of your iterative algorithm?

O(n,m)

a grid of numbers, find maximum length Snake sequence and print it. If multiple snake s exists with the maximum length, print any one of them. A snake sequence is made up of it numbers in the grid such that for each number, the number on the right or the number below it is -1 its value. For example, if you are at location (x, y) in the grid, you can either move right i.e. (x, ) if that number is ± 1 or move down i.e. (x+1, y) if that number is ± 1.

For example,
9, 6, 5, 2
8, 7, 6, 5
7, 3, 1, 6
1, 1, 1, 7
In above grid, the longest snake sequence is: (9, 8, 7, 6, 5, 6, 7) with path length 7. [3 + 2+ 5 = 10 Marks]

Below figure shows all possible paths:




r rows
c columns

(a) Design recursive equation for solving this problem

FN(r,c)
IF (r=1 AND c=1) return 0
~~ELSE IF ((r≠0 AND) (c≠0))~~
ELSE IF (c=1 AND |A[r,c] - A[r-1,c]| = 1) RETURN FN(r-1,c)+1
ELSE IF (c=1 AND |A[r,c] - A[r-1,c]| ≠ 1) RETURN 1
ELSE IF (r=1)
IF (|A[r,c] - A[r,c-1]| = 1) RETURN FN(r,c-1)+1
ELSE RETURN 1
ELSE IF (|A[r,c] - A[r,c-1]| = 1 AND |A[r,c] - A[r-1,c]| = 1)
RETURN MAX (FN(r,c-1), FN(r-1,c)) + 1
ELSE IF (|A[r,c] - A[r,c-1]| = 1) RETURN FN(r,c-1) + 1
ELSE IF (|A[r,c] - A[r-1,c]| = 1) RETURN FN(r-1,c) + 1
ELSE RETURN 1

(b) Write iterative dynamic programming pseudocode.

```
DP[1..r, 1..c]
FOR(i = 1 to r)
    FOR(j = 1 to c)
        IF (i = 1 AND j = 1)        DP[i,j] = 1
        ELSE IF (i = 1)
            IF ( |A[i,j] - A[i,j-1]| = 1 )
                DP[i,j] = DP[i,j-1] + 1
            ELSE    DP[i,j] = 1
        ELSE IF (j = 1)
            IF (|A[i,j] - A[i-1,j]| = 1 )
                DP[i,j] = DP[i-1,j] + 1
            ELSE    DP[i,j] = 1
        ELSE IF (|A[i,j] - A[i,j-1]| = 1 AND |A[i,j] - A[i-1,j]| = 1)
            DP[i,j] = 1 + MAX( DP[i-1,j], DP[i,j-1])
        ELSE IF (|A[i,j] - A[i-1,j]| = 1 )
            DP[i,j] = DP[i-1,j] + 1
        ELSE IF (|A[i,j] - A[i,j-1]| = 1 )
            DP[i,j] = DP[i,j-1] + 1
        ELSE    DP[i,j] = 1
```

(c) What is time complexity of dynamic programming algorithm?

$O(rc)$

two arrays A and B of equal size n, you have to design an efficient algorithm that
s the sum A[1] x B[1] + A[2] x B[2] + … + A[n] x B[n]. You are allowed to shuffle the
nts of each array, A and B. [7 Marks]

*mple:*

*=3*

*A = {3, 1, 1}*, B = (6, 5, 4)

*Minimum* sum = 1 x 6 + 3 x 4 + 1 x 5 = 23

*There* are other possible ways of taking the sum like 3 x 6+ 1 x4+1 x5 = 27, but the minimum sum is 23.

***Hint:* Use greedy algorithm**

Sort A in increasing order

Sort B in decreasing order