National University of Computer and Emerging Sciences, Lahore Campus

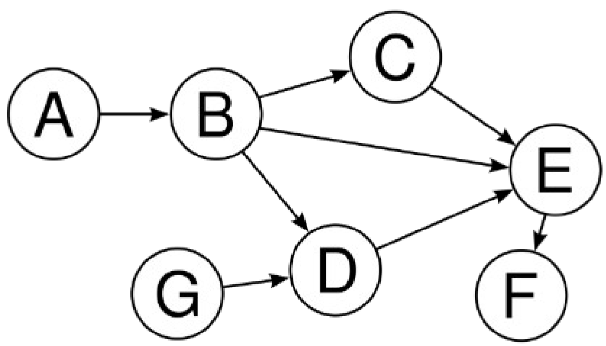
Course:	Design and Analysis of Algorithms	Course Code:	CS302
Program:	BS(Computer Science)	Semester:	Spring 2018
Duration:	60 Minutes	Total Marks:	23
Paper Date:	13-April-18	Weight	15%
Section:	ALL	Page(s):	4
Exam:	Midterm 2 Solution		

Instruction/Notes: Attempt the examination on the question paper and write concise answers. You can use extra sheet for rough work. Do not attach extra sheets used for rough with the question paper. Don't fill the table titled Questions/Marks.

Question	1-4	5	6	Total
Marks	/ 10	/ 8	/ 5	/ 23

Question#1

What would be the output (ordering of vertices) of topological sort on the following graph, starting with vertex A. [4 Marks]



Answer: G A B D C E F

Question#2

What would be the time complexity of Breadth First Search in the worst case, if graph is represented as Adjacency Matrix instead of Adjacency List? [2 Marks]

O (n^2)

Question#3

Given an adjacency list of a directed graph, how long will it take to compute the outdegree of each vertex? Also how long would it take to compute the indegree of each vertex? [2 Marks]

Answer: O (m + n)

Question#4

Six files F1, F2, F3, F4, F5 and F6 have 100, 200, 50, 80, 120, 150 records respectively. In what order should they be stored so as to optimize file access (Access time of file depends on number of records stored before this file). Assume each file is accessed with the same frequency. [2 Marks]

a) F3, F4, F1, F5, F6, F2
b) F2, F6, F5, F1, F4, F3
c) F1, F2, F3, F4, F5, F6

Answer: option a

<u>Question#5</u>

Below is the recurrence relation for edit distance of strings X_n and Y_m where $X_n = (x_1, x_2, ..., x_n)$ is a string of n characters and $Y_m = (y_1, y_2, .., y_m)$ is a sting of m character:

$$ED(X_n, Y_m) = \begin{cases} m \; if \; n = 0 \\ n \; if \; m = 0 \\ ED(X_n - 1, Y_m - 1) \; if \; x_n = y_m \\ Min(ED(X_{n-1}, Y_{m-1}), ED(X_{n-1}, Y_m), ED(X_n, Y_{m-1})) + 1 \; if \; x_n \neq y_m \end{cases}$$

Typists often make transposition errors exchanging neighboring characters, such as typing “setve” when you mean “steve.” This requires two replace operations (edit distance=2) to convert setve to steve. If we incorporate a swap(swaps two adjacent characters) operation into our edit distance functions along with insert, delete and replace, then such neighboring transposition errors can be fixed at the cost of one operation.

a) What would be the recurrence relation for this modified edit distance (that includes swap operation)? [4 Marks]

$$ED(X_n, Y_m) = \begin{cases} m \; if \; n = 0 \\ n \; if \; m = 0 \\ ED(X_n - 1, Y_m - 1) \; if \; x_n = y_{m\text{¿}} \text{¿} \, ED(X_n - 2, Y_m - 2) + 1 \; if \; x_n \neq y_m \wedge x_{n-1} = y_m \wedge x_n = y_{m-1} \\ Min(ED(X_{n-1}, Y_{m-1}), ED(X_{n-1}, Y_m), ED(X_n, Y_{m-1})) + 1 \; if \; x_n \neq y_m \end{cases}$$

b) Write down the pseudo code of a function that computed the edit distance of X_n and Y_m [3 Marks]

Answer:

```
ED (X, Y)
{
    for (i=0   to n)
    {
        For ( j=0 to m )
        {
            If ( i =0)  EditDistance[i][j]  = j
            Else If ( j =0)  EditDistance[i][j]  = i
            Else if ( Xn = Ym )    EditDistance[i][j]  = EditDistance[i-1][j-1]
            Else if ( Xn-1 = Ym  and Xn = Ym-1)    EditDistance[i][j]  = EditDistance[i-2][j-
2]  + 1
            Else EditDistance[i][j]  = Min (EditDistance[i-1][j-1] ,    EditDistance[i-1][j],
EditDistance[i][j-1] ) +1

        }
    }
}
```

c) What is the time complexity of your algorithm? [1 Mark]

O (m n)

Question#6

Consider the weighted interval scheduling problem. Here the input is a set of n jobs, each with a start time finish time, and reward. Our task is to schedule some of these jobs on a single machine. The output is a non-overlapping subset of the jobs, and the goal is to maximize the total reward from the jobs in the set.

Consider following greedy strategy:

Sort the jobs by reward and schedule them one by one starting with the highest reward, rejecting any that overlap with jobs already scheduled.

Either prove that this greedy strategy always gives optimal solution or give counter example to prove that it does not guarantee optimal solution. [5 Marks]

Solution:

Counter Example:

Job	1	2	3
Start	1	2	4
Finish	7	3	5
Reward	50	30	30