Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programmes Division
First Semester 2012-2013
Comprehensive Examination (EC-3 Regular)

Course No.	: SS ZG519
Course Title	: DATA STRUCTURE AND ALGORITHMS DESIGN
Nature of Exam	: Open Book
Weightage	: 50%
Duration	: 3 Hours
Date of Exam	: 04/11/2012 (AN)

No. of Pages = 2
No. of Questions = 3

Note:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. Assumptions made if any, should be stated clearly at the beginning of your answer.
3. Start each question from a fresh page. Make suitable assumptions wherever required, and explicitly mention them in the beginning of each question. Be precise and concise while answering the questions.

Q.1. Suggest suitable data structures for modeling each of the following problems (separately):

(a) Back operation in windows explorer or other graphical directory browser. [Assume that multiple browser windows cannot be opened.] 9

(b) searching contact by name or number in a smart phone. [Assume unlimited space for storing the contacts.]

(c) Determining top-10 gainers in stock market on a given day.

[Note: Provide only data structures to be used. No algorithm should be written for this question.] [3*2 = 6]

Q.2. Consider the problem defined as follows:

You are traveling by a canoe down a river and there are n trading posts along the way. Before starting your journey, you are given for each $1 \leq i < j \leq n$, the fee $f_{i,j}$ for renting a canoe from post i to post j. These fees are arbitrary. For example it is possible that $f_{1,3} = 10$ and $f_{1,4} = 5$. You begin at trading post 1 and must end at trading post n (using rented canoes). Your goal is to minimize the rental cost. Be sure to prove that your algorithm yields an optimal solution and analyze the time and space complexity.

[Hint: You can try defining a metric m[i] to be the rental cost for the best solution to go from post i to post n for $1 \leq i \leq n$. The final answer is in m[1]. The canoe must be rented starting at post i (the starting location) and then returned next at a station among i + 1, . . . , n. In the recurrence, all possibilities (with j being the station where the canoe is next returned) must be tried.

$$m[i]=\begin{cases} 0 & \text{if } i=n \\ \min\limits_{i<j\leq n}\left(f_{i,j}+m[j]\right) & \text{otherwise} \end{cases}$$ End Hint.]

(a) To which complexity class does this problem belong to?

(b) By using Memoization

(c) By using Dynamic Programming

(d) By using any greedy method

(e) If instead of restricting that the best path is always in the forward direction (path defined as p1, p2,..,pi, pj,...,pk, such that i<j), it is stated that the best path may involve any ordering of intermediate stations, then what will be the change in running time for the algorithms suggested by you, and why?

[2+ 3*6 + 6 = 26]

q. (b)

Q.3. Given a traffic network containing cities and roads connecting them, it is desired to compute the following:

(i). Verify whether all the places are reachable by road from every other place.
(ii). Determine all places within x km distance from a given place
(iii). To find disjoint paths between a city and two other cities.

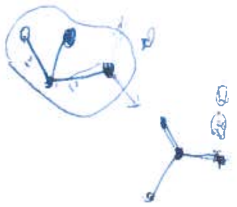
(a) Suggest a mechanism to model the problem into graph. Explicitly state whether you would choose a directed or undirected graph with proper justification.

BFS DFS

(b) Write an algorithm/pseudo-code to verify whether all cities are reachable by road from every other city.

(c) Write an algorithm/pseudo-code to determine all places within x km distance from a given place A by exploring level by level on the basis of closeness of the places connected to A.

Dijkstra Prim (A)

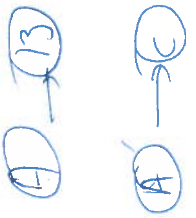


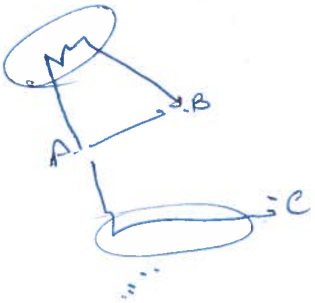
(d) Given three places A, B and C in the traffic network, determine two disjoint paths between A->B and A->C such that there are no roads common (no two edges are common) to the routes A->B and A->C . [Hint: Modify the DFS algorithm appropriately to avoid common paths between A->B and A->C, by avoiding the edges already marked as common.] If such disjoint paths are not possible, then the algorithm should return an error.

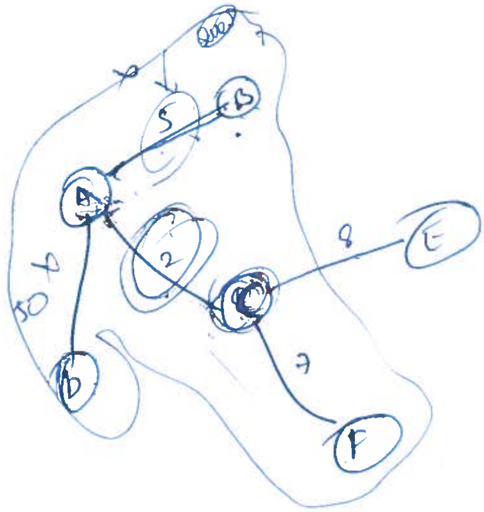
[3 + 3 + 4 + 8 = 18]

/DFS(A,Q,n.

****







A B - 5 ✓
A-B