**Leading University, Sylhet**



**Department of Computer Science & Engineering**

**Semester Final Exam, Summer-2016**

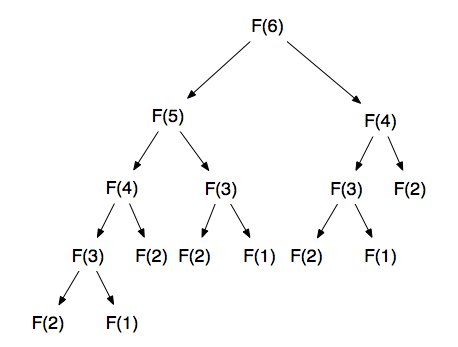
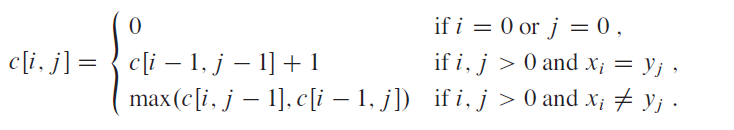
**Course Title: Computer Algorithms and Complexity Course Code: CSE-2117**

**Full Marks: 40 Time: 2:00 Hours**

**Answer any 4(4x10=40) questions from the following. [Each question carries equal marks]**

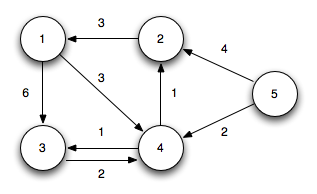
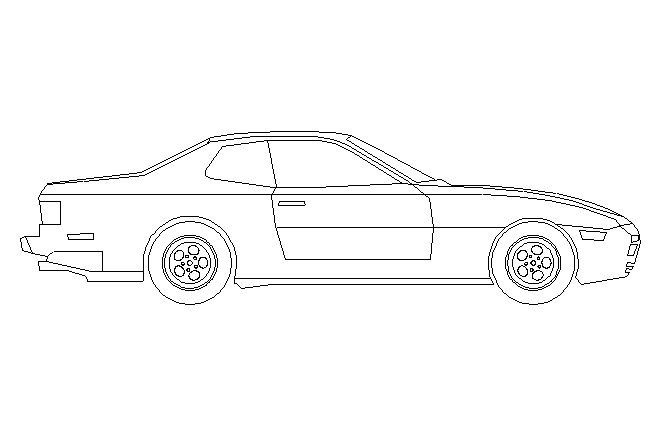
1. a) Write down the differences between Dynamic Programming and Greedy Algorithm.  **[2]**

b) Following diagram is the recursion tree to compute the 6th Fibonacci number using recursion.   
 Do you think Dynamic programming is applied here? If so, then explain how is dynamic   
 programming applied here. If not, then explain why?

 **[2]**  
 c) Following is the recursive algorithm to compute **LCS** of two string **x** and **y**. Do you think   
 Dynamic Programming is applied here? If so, then explain how?   
 If not, then explain how Dynamic Programming can be applied here?  
 Here **i** is the index of string **x** and **j** is the index of string **y**. **C** is a 2-d array where length of   
 LCS of index (**i**,**j**) of the string **x** and **y** is stored.   
   **[2]**  
 d) Explain with an example why optimal parenthesization is important for matrix chain   
 multiplication. Explain for the multiplication of A1.A2.A3 with dimensions 10 × 100,   
 100 × 5, 5 × 50.  **[3]**

e) What are the two approaches of Dynamic Programming?  
 **[1]**

1. a) You are developing a Racing Game. Now you want to determine all the shortest route for the **AI** **car** of your game. In the following graph, there are 5 vertexes numbered 1,2,3,4 and 5. These are the 5 destinations that the **AI car** can go. Here the **source** is the vertex/node with number **5** where the **AI car** is located initially. Now apply **Dijkstra algorithm** in the following graph to find all the **shortest distance** from the **source**. You need to write each steps.

**[5]**

b) State the *advantages* and *disadvantages* of **Dijkstra** and **Bellmen ford** algorithm.

**[2.5]**

c) Following is the pseudocode of **Breadth-First-Search Algorithm** which has some error. Identify the error in the code and provide the solutions **or** rewrite the whole code so that it would be error free.

**Breadth-First-Search (Graph, root):**   
 **for** each node n in Graph: // Graph is an array consisting of all the nodes

n.distance = INFINITY // set the distance of all nodes to infinity  
n.parent = NIL // set the parent of all nodes to NIL

**end for**

create empty **stack** S   
 root.distance = 0

root.parent=root  
 **S**.push(root) // push root in the stack

**while** S is not empty: // run the while loop till it is not empty

current = S.top() // top element of stack is stored in current  
 S.pop() // pop the top element of stack  
   
 **for** each node **n** that is adjacent to current:   
 **if** n.distance != INFINITY:

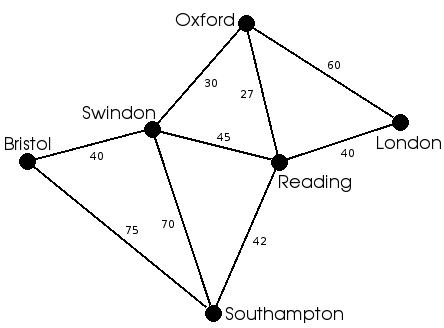
n.distance = current.distance // set the distance of node **n**   
n.parent = n // set parent of node **n** as **n**

S.push(n) // push node **n** in the stack

Breadth-First-Search(Graph,current) // recursive call  
 **end if**  
 **end for**  
 **end while**

**[2.5]**

1. a) Following Graph contains the road network in Southern England. Now find the **Minimum Spanning Tree(MST)** of the following Graph using Kruskal’s MST Algorithm. Each edge contains the weight between the adjacent areas. Each vertex contains the name of the area.



**[4.5]**

b) How is the time complexity of **Prim’s Algorithm** O(ELogV) where **E** indicates edges and **V** indicates vertices.

**[2.5]**

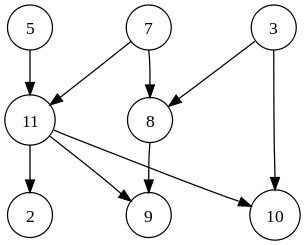
c) Is *Backtracking* applied in DFS? If so, then explain with an example.

**[2]**

d) Select the appropriate data structure of **Breadth First Search**, **Depth First Search**, **Prim's Minimum Spanning Tree,** **Kruskal's Minimum Spanning Tree** from the following.   
  
 i) Priority Queue ii) Queue iii) Stack iii) Union Find

**[1]**

1. a) Apply **Topological Sort** in the following graph. The number inside each vertex is the node number of that vertex. You have to write each steps.

  
 **[5]**

b) Calculate the length of **Longest Increasing Subsequence(LIS)** of the following sequence   
 using bottom up DP table.

9,7,1,4,99,71,49 **[2]**

c) Describe the following operations of **Union Find Disjoint Set**.   
  
 i) MAKE-SET(x) ii) UNION(x,y) iii) FIND-SET(x) **[3]**

1. a) **Huffman coding** is a lossless data compression algorithm. You have a text file in your computer which you want to compress.   
   The text file contains the following text “abccddbdcdeeffeeeffff”. Now Apply **Huffman encoding** in that text.   
     
   You have to do the following task to Apply **Huffman Encoding**:  
     
    i) Construct the Huffman Tree from that given text using frequency of each characters.

ii) Generate the binary code of each characters of that string from the Huffman Tree.  
  **[6]**

b) Suppose you are working as Software Engineer at Facebook. You have to sort billions of data there. As a Software Engineer you have to do the task in the most optimized way so that in worst case your selected algorithm performs well. You know many sorting algorithms like **Insertion Sort, Bubble Sort, Quick Sort, Merge Sort, Heap Sort, Selection Sort** etc. In such a case where you have to sort billions of data, which of the above **algorithms** would you prefer? Justify your answer.

**[2]**

c) What is priority queue? Do you use priority queue in **Bellmen ford algorithm**? If yes, then explain how it is used in Bellmen ford algorithm, if not then name a **Single Source Shortest Path(SSSP)** algorithm where priority queue is used.   
   
 **[2]**