The Open University of Sri Lanka
Bachelor of Software Engineering
Department of Electrical and Computer Engineering
ECX 4265 – Data Structures & Algorithms
Final Examination 2013/2014

Date: 30th August 2014	**Time: 0930 - 1230hrs**

This paper contains Eight (08) questions on five pages. You need to **Answer FIVE questions ONLY.**

Q1.

Consider the scenario of a trapped mouse trying to exit a maze as shown in Figure 1. The maze is implemented as a two dimensional character array in which passages are marked with 0s, walls by 1s, exit position by the letter ***e***, and the initial position of the mouse by the letter ***m*** as shown. The mouse tries to exit the maze systematically trying all possible routes; first it tries the upper neighbour, then lower, then left and finally right. Each possible route will be stored in a stack with respect to the current position of the mouse. Then mouse takes the top most position of the stack as the route and proceed towards the exit.

1 1 0 0
0 0 0 e
0 0 m 1
Figure 1

(i) Write a pseudo code algorithm for the above scenario of the mouse escaping the maze. (12 marks)

(ii) "Time complexity is more important than Space Complexity" comment on this statement. (2 marks)

(iii) Explain why Worst case time complexity calculation is more realistic than Best case or Average case time complexity. (2 marks)

(iv) Compute the time complexity of the following with respect to Big oh notation.

```
i =1;
while (i<20)
{
        for (j=0; j < n; j++)
                for (k=0; k < j; k++)
                {
                        actions.....
                }
i++;
}
```

(4 marks)

Q2.

(i) Compare and contrast non linear data structures with linear data structures by giving examples for each. (3 marks)

(ii) Consider the data set given below

12, 3, 45, 6, 7, 11, 4, 47, 12, 2, 90, 20

(a) Write a pseudo code algorithm for arranging the data set into ascending order using a suitable sorting algorithm. (6 marks)

(b) Write a pseudo code algorithm to insert the above sorted data set into a stack data structure. (6 marks)

(c) A Binary search tree is built by inserting the data in the stack data structure in part(b). Draw the output binary search tree. (5 marks)

Q3.

(i) Describe the ***Open Addressing*** and ***Separate Chaining*** methodologies used for collision resolution. (4 marks)

(ii) Insert the following numbers to a one column table of size ten (10) according to below given collision resolving methods.

55, 77, 167, 100, 357, 858, 346, 201, 535, 22

(a) Use linear probing to resolve collisions,
(b) Use separate chaining to resolve collisions. (4 marks each)

(iii)

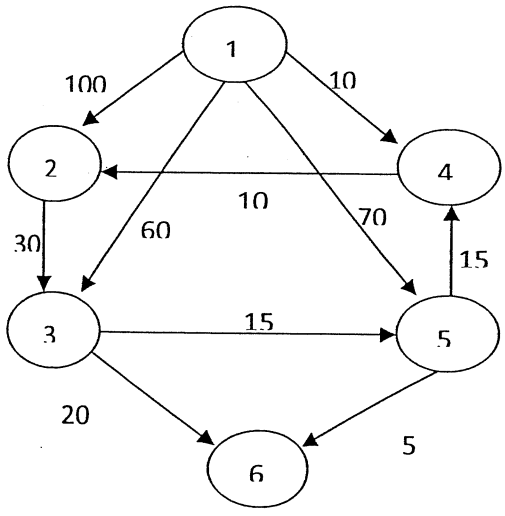

Figure 2

(a) Use Dijkstra's shortest path algorithm to calculate the shortest path from vertex 1 to each vertex of the graph shown above. Show the predecessor and distance arrays following each step of Dijkstra's algorithm. (8 marks)

Q4.

(i) Delimiter matching is part of any compiler and is a good example for application of stack data structures. Assume that we have to match the following delimiters parentheses "(" and ")", square brackets "[" and "]", curly brackets "{" and "}", and comment delimiters "/" and "\".

(a) Write a pseudo code algorithm for matching delimiters. (6 marks)

(b) Hand trace how the above written algorithm will work on the below expression

$$s = t[5] + u/ (v * (w + y))$$ (5 marks)

(ii) Briefly explain what data structure can be used to implement a hospital emergency room schedule; an operating table. (2 marks)

(iii) Assume that a critically injured person was brought to the hospital and he has to be given the top priority. Write a pseudo code algorithm to insert the patient at the first position based on the data structure you suggested in part (ii). (7 marks)

Q5.

(i) Describe how merge sort can be used when the size of the dataset to be sorted is larger than the available memory. (2 marks)

(ii) Write a pseudo code algorithm for merge sort. (4 marks)

(iii) Execute the above written merge sort algorithm on the array B={3, 10, 8, 6, 12, 7, 5, 4, 24}. You need to illustrate all intermediate values of the array. (7 marks)

(iv) To avoid doubling of the workspace needed when arrays are sorted with merge sort, it is better to use a linked list of data instead of an array. Write a pseudo code to implement merge sort with a set of data stored in a linked list. (7 marks)

Q6.

(i)
```
function pigeonhole_sort(array a[n])
        array b[N]
        var i,j
        zero_var (b)     (* zero out array b *)

for i in [0...length(a)-1]
        b[a[i]] := b[a[i]]+1

        (* copy the results back to a *)
        j := 0
for i in [0...length(b)-1]
        repeat b[i] times
        a[j] := i
            j := j+1
```

(a) Number the steps in the above procedure and explain what is happening at each step. (5 marks)

(b) Describe how you can apply the above algorithm into the data set A={ 1, 4, 2, 5, 4} (4 marks)

(ii) The pigeonhole sort used in part (i) is not efficient as quick sort. Write a complete pseudo code algorithm for quick sort. (4 marks)

(iii) Hand traces the following set of numbers given in the array using quick sort. Assume the pivot as the first element.

int C = { 9, 1, 12, 15, 4, 18, 13, 6, 2} (5 marks)

(iv) Briefly describe why quick sort is better than pigeonhole sort for large data sets. (2 marks)

Q7.

(i) Insert the following dataset in to a binary search tree according to the given order.

50, 24, 87, 12, 6, 28, 74, 66, 92, 4, 80, 97, 55, 32 (5 marks)

(ii) Write a pseudo code algorithm for in-order traversal of the binary search tree and give the output of the in-order traversal. (6 marks)

(iii) Write a pseudo code algorithm for deleting a non leaf node from a binary search tree. (5 marks)

(d) Write the output of the pre-order traversal and post order traversal of the binary search tree you build in part (i). (4 marks)

Q8.

(a) Below given is a directed graph.

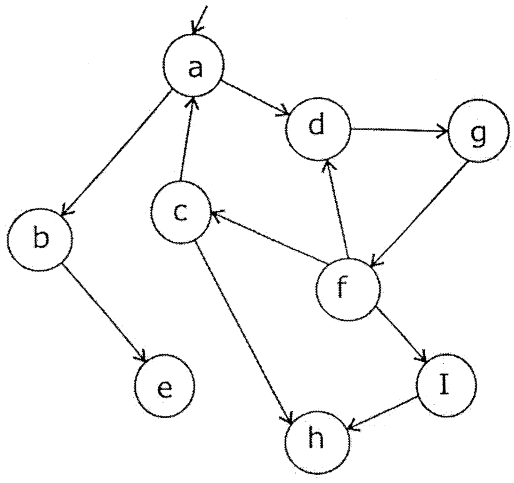

Figure 3

(i) Draw the corresponding adjacency list representation of the above graph. (5 marks)

(ii) Another way to represent a graph is an adjacency matrix. Draw the adjacency matrix for the above graph. (3 marks)

(iii) List the *breadth first* and *depth first* search sequences for the above graph. Assume the search starts at node 'a'. (6 marks)

(iv) Describe the following types of graphs

(a) Directed graphs
(b) Weighted graphs
(c) Unconnected graphs (2 marks each)

The End