AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
Computer and Systems Engineering Department
3rd Year Computer and System Engineering Students

2nd Semester, 2013-2014 **Course Code: CSE323** ***Time : 3.00 Hrs***

Data Structures and Their Algorithms

The exam consists of **six** Questions in **Three** Pages. ***Total Marks: 90 Marks*** **1/**7

Answer all Questions

Question 1.

State true or false and justify your answer. **[10 marks]**

a) Any (descended order) sorted array is considered a heap.
(true) because each value is smaller than its parent which comes before it

b) A complete binary tree of depth n has 2n nodes
(false) it is the sum of: 1 ,2 , 4,…., 2^n which is (2^(n+1) -1) in case of depth 3 the number of nodes are 15

c) The big O of insert in a binary search tree is log(n) on average
(true) average case is a complete tree so the distance from root to any leaf node is log(n)

d) In some cases, a Binary search algorithm does better than binary search tree
(true) Binary search algorithm is better in case of not balanced tree

e) Selection sort has better big O than merge sort
(false) merge is n log(n) where selection is n^2

f) Heap sort is better than merge sort memory space-wise
(true) it doesnot require other arrays

g) Bubble sort has more swaps than selection sort
(true) number of swaps has order n^2 in selection they are just n

h) The minimum element in a heap is always a leaf node
(true) to satisfy the heap property

i) Removing the last node in a linked list has big O of log(n)
(false) it is O(n)

j) The maximum element in a binary search tree is always a leaf node
(false) it does not have right child but may have a left child

Question 2.

Given a sorted (ascending order) linked list containing positive and negative numbers with their frequencies (repetition) as follows: →(-9 , 2) →(5, 7) →(7, 3) →(9, 2) →(11, 1) The first number is the value and the second is its repetition (frequency) for example, in the first node (-9 is the value and 2 is its repetition) a- Write a C++ class for the node containing all required data and methods **[4 marks]**

```
class node
{int v, f; //value and frequency
node * next;
public:
int getv() {return v;}
int getf() {return f;} }
```

b- Write a C++ class for the list or chain that contains the whole list of nodes with all data and the prototypes of the methods **[4 marks]**

```
class list{
node * chain;
public:
list() {chain = null;} //constructor
void insert(node n);
int getaverage();
}
```

c- Write C++ methods of class list to perform the followings:

- find the average value of the positive numbers **[4 marks]**

  a simple loop where the product of positive values and frequency is computed and divided over the sum of the frequencies

```
int list::getaverage()
{node *temp=chain;
Int s=0, t=0;
while(temp!=null)
{ if(temp->getv()>0)
{t=t+temp->getf();
s=s+temp->getf()*temp->getv();}
temp =temp->next;}
return s/t;

}
```

- insert a new node to the list ; list must kept sorted [4 marks]

```
void insert(node n)
{node *temp=chain;
While (temp !=null)
If(n.getv()>=temp.getv() && n.getv()<=temp->next->getv())
//insert node
Temp=temp->next; }
```

Question 3.

Given the following recursive code

```
void recur(stack s)

{ if(! s.isempty())

{x=s.pop();

cout<<x<<"\t";

recur(s);

s.push(x);}}
```

a) What is the stopping condition of this recursive function **[4 marks]**
Empty stack will stop the recursion

b) Trace the function when it is called with a stack containing integer numbers [10, 3, 26, 5[; number 10 is at the bottom and number 5 is at the top **[6 marks]**
Print 5 then 26 then 3 then 10 at the end the stack will remain as before containing the four values

c) What is the big O w.r.t n (the number of items inside the stack)? **[6 marks]**
Big O here is n there are n print, n push and n pop

Question 4.

Given two heaps of integers like [15 9 8 7 5 3] and [10 6 4 2]

Data Structures and Their Algorithms

a) Write down three C++ functions to return the left child, right child, parent of a given node. Your function input is an index and the result/returned value is an index. **[3 marks]**

```
int lchild(int n) {return n*2+1;}
int rchild(int n) {return n*2+2;}
int parent(int n) {return (n-1)/2;}
```

b) Merge the two heaps into a single heap of 10 elements **[3 marks]**

Simply add them as you are sorting two sorted arrays

[15 9 8 7 5 3] and [10 6 4 2] → [15 10 9 8 7 6 5 4 3 2]

c) Write an algorithm or C++ function to merge them **[4 marks]**

same as merge two arrays in the mergesort

c) What is the big O of your algorithm in (b) with respect to m and n the number of items in both heaps? **[3 marks]**

O(m+n)

d) Remove the item 15 from the first heap and show step by step how the heap is reorganized. **[3 marks]**

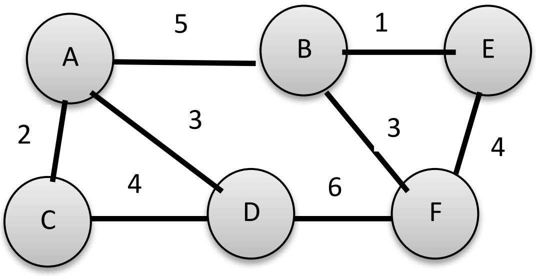
[15 9 8 7 5 3] → [3 9 8 7 5] – reheap →[9 3 8 7 5] – reheap →[9 7 8 3 5]

Question 5.

Given is an undirected graph of 6 vertices.

assume the nodes in the adjacency matrix or list

appear in their alphabetical order.



a) Is this graph sparse or dense? **[2 marks]**

Sparse n =6 e=8

b) How many edges are required to make it complete? **[2 marks]**

Complete graph undirected has (6*(6-1))/2= 15 we have e=8 so we need 7

c) Show how this graph's edges can be represented/stored in memory using adjacency matrix. **[4 marks]**

	A	B	c	d	e	F
A		5	2	3		
B	5					
C	2					
D	3					
E						
F						

Etc.

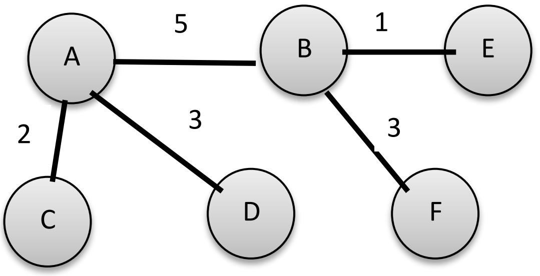
d) Write an algorithm to determine whether an undirected graph is connected or not (a connected graph means you can reach any node starting from any other node) apply your algorithm on the shown graph. **[4 marks]**

Similar to breadth first algorithm at the end we check if one of the nodes is not marked the graph is disconnected (We may start from any node)

e) Traverse the graph (breadth first). Show the order of the vertices (assume marking is done when the vertex is placed inside the queue or the stack) **[4 marks]**

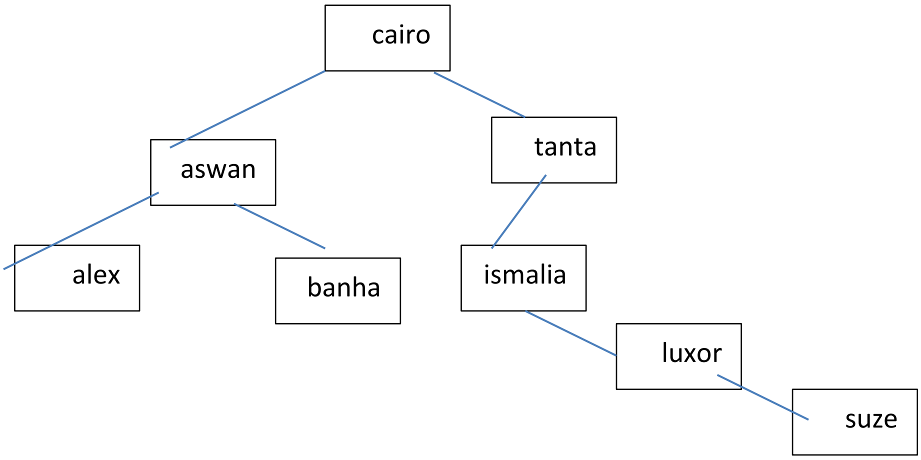
A, B, C, D, E, F

f) Find the minimum spanning tree of this graph starting from vertex D **[4 marks]**

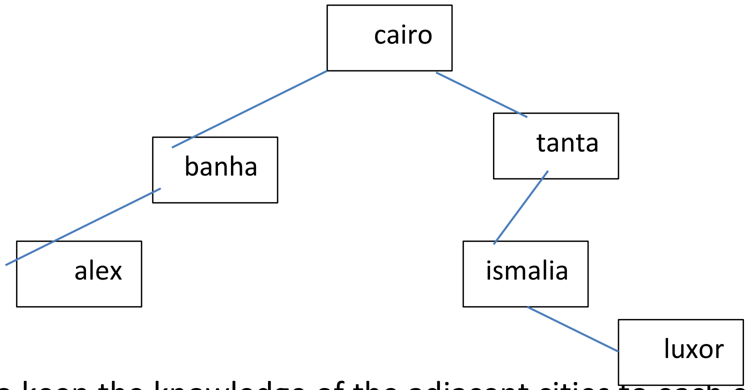


Question 6.

a) Insert the following city names into a binary search tree, where they will be ordered alphabetically {"cairo", "aswan","alex","tanta","banha","ismailia", "luxor", "suze"} **[4 marks]**

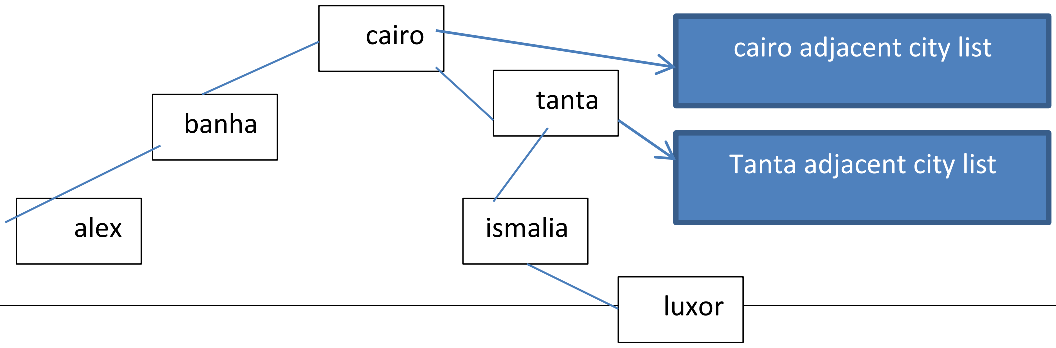


b) Show the tree after deleting "aswan" and "suze". **[4 marks]**



c) If it is required to keep the knowledge of the adjacent cities to each city within the binary search tree, show how it may be represented and find the big O of finding the adjacent cities of a given city **[4 marks]**

we would have a linked list merging out from each city in the BST



the big O would be log(n) to find the city followed by m number of cities adjacent to the given city so the overall could be O(n)

Good Luck