**Do the following Activities and submit**

**TREE**

1. What is the difference between a tree and a binary tree?

2. Draw an expression tree for each of the following expressions:

(a) 5

(b) (5+6\*4)/2

(c) (5+6\*4)/2-3/7

(d) 1+9\*((5+6\*4)/2-3/7)

(e) A × B − (C + D) × (P /Q)

3. Hand draw binary expression trees that correspond to the expressions for which

(a) The infix representation is P /(Q + R) ∗ X − Y

(b) The postfix representation is X Y Z P QR ∗ +/ − ∗

(c) The prefix representation is + ∗ −M N P /RS

4. For each expression tree drawn in the above question, list the sequence of elements encountered in inorder, preorder and postorder traversals.

5. Describe, with an example of 5 nodes, the topological characteristics that distinguish a tree from a linked list.

6. Define a treeNode class for tree nodes that each consists of following 4 fields:

|  |  |  |  |
| --- | --- | --- | --- |
| parent | leftChild | rightChild | data |

7. Following the above node definition, define and implement a binary tree class with necessary access methods for the 4-field treeNodes, for example, setXXX, getXXX, isEmpty, etc..

8. Write a method that takes two binary trees t1, t2 and a binary tree node v as the arguments. It constructs and returns a new binary tree that has v as its root and whose left subtree is t1 and whose right subtree is t2. Both t1 and t2 should be empty on

completion of the execution.

**H E AP S**

1. What is the difference between a binary tree and a tree in which each node has at most two children?

2. What is the approximate number of comparisons of keys required to find a target in a complete binary tree of size n?

3. What is a heap?

4. What is a priority queue?

5. Demonstrate, step by step, how to construct a max-heap for the list of integers A[1..8] = (12, 8, 15, 5, 6, 14, 1, 10) (in the given order), following Example 4.14.

6. Demonstrate, step by step, how to sort a list of integers (2, 4, 1,

5, 6, 7) using a max-heap, following Example 4.15.

GRAP H S

1. Consider the adjacency matrix of a graph below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |

A =

(a) Draw the graph

(b) Write the adjacency list for the graph

(c) Discuss the suitability of using an adjacency matrix and the adjacency list for the graph. Justify your answer.

2. Using the adjacency matrix approach, write a program to store a simple graph and display the graph.

**Example 4.23**

Store and Display a simple graph

-------------------------

1. Store a graph

2. Display a graph

0. Quit

Please input your choice (0-2) >

**Hint** An easy approach may be:

(a) define a data structure for the graph

(b) decide a means to input the graph, for example, you may

i. type the entries of the adjacency matrix on the keyboard ii. read the entries of the adjacency matrix from a text file iii. generate a random adjacency matrix by a program.3

3

Use a random generator to generate

a 0 or 1 uniformly at random for each entry of the matrix.

(c) write the main program or method with interfaces of the

sub-methods or procedures

(d) develop each part of the program.

3. Implement the graphs in question 1 using the adjacency list

approach.