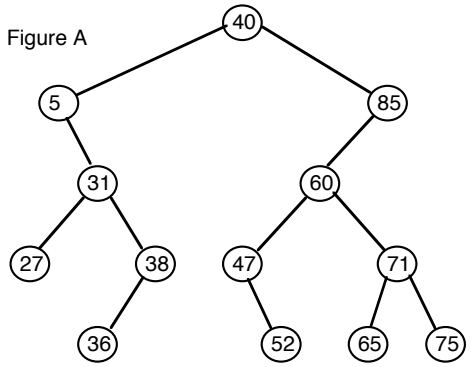
Fundamentals of Data Structures
Trees
Example test questions for the course

These questions or similar questions have been used in tests in previous years. The course has been taught using various program languages, as a consequence the program text in these examples may be written in Java, Pascal, Turing or pseudo code.

---

1. Remove the entry with key 40 from the binary search tree in Figure A. Draw the resulting tree.

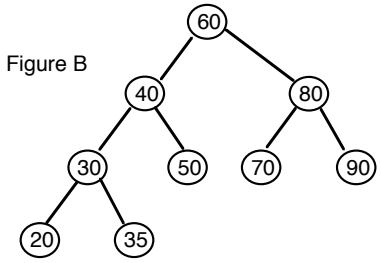
Figure A



---

2. Insert an object with key value 10 into the AVL tree in Figure B. Draw the rebalanced AVL tree after the insertion.

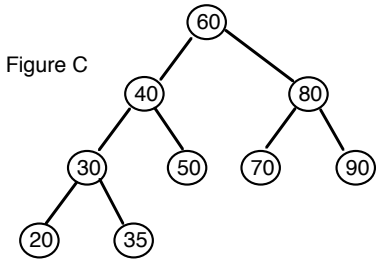
Figure B



---

3. Remove the object with key value 50 from the AVL tree in Figure C. Draw the rebalanced AVL tree after the removal.

Figure C



4. Insert an object with key value 38 into the AVL tree in Figure C. Draw the rebalanced AVL tree after the insertion.

5. Give a mathematical expression that describes the heap property for a binary tree. Describe what your expression means in a few English sentences.

6. Give a mathematical expression that describes an AVL tree. Describe what your expression means in a few English sentences.

7. Give a mathematical expression that describes an AVL tree. Describe what your expression means in a few English sentences.

8. Give a mathematical expression that describes the heap property for a binary tree. Describe what your expression means in a few English sentences.

9. Define formally the heap data structure.

10. Given an AVLnode as defined in Assignment 3. Complete, in psuedocode the following algorithm.

    Require: A ~= null **and** B ~= null
    Ensure: Result = True ↔ A is an ancestor of B **or** B is an ancestor of A

```
boolean anscestorRelated(
    AVLnode root,           // root of the tree to search
    AVLnode A, AVLnode B    // Potentially related nodes)
is
```

    What is the Big O for the running time?

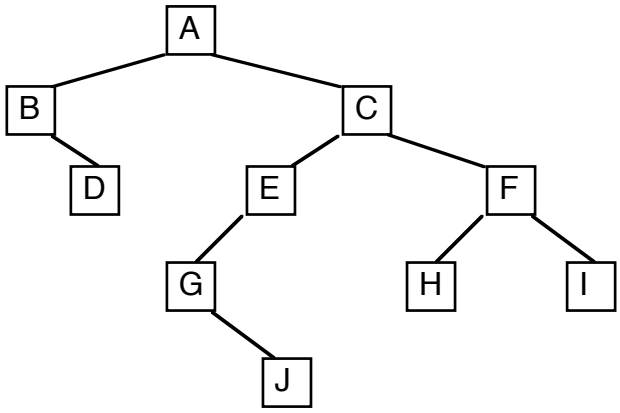
11. Complete the following pseudocode recursive algorithm to determine if a given array contents has the heap property. Larger integers should be nearer the root in the tree representation.

    Require: True
    Ensure: Result = True ↔ A[lb .. ub] has the heap property as described above

```
boolean isHeap(int A[],    // The array
               int lb,     // The lower bound
               int ub      // The upper bound  )
{
```

    What is the Big O for the running time?

12. Given the following tree.



List the nodes of the above binary tree in the following orders.

Inorder ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

Preorder ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

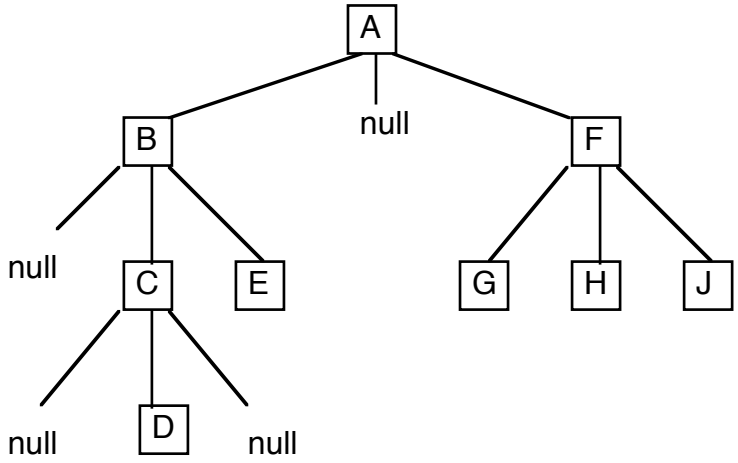
Postorder ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

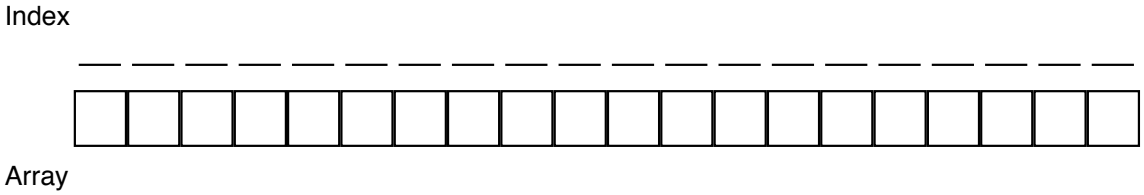
13. Draw a series of heap tree diagrams as elements are added to and removed from a priority queue that is implemented as a heap. Higher numbers indicate higher priority. Add and remove elements in the given order. The initial heap is empty.

Add 10, Add 20, Add 30
Add 40, Remove 1 element, Add 70
Add 25, Remove 1 element, Add 35
Add 28, Remove 1 element, Remove 1 element
Add 50, Add 55, Add 21
Remove 1 element, Remove 1 element, Remove 1 element

14. Write down the equations that define the relationship between the parent and child nodes in a ternary tree (degree 3 tree) when using array representation with all array elements representing valid nodes notation, with no gaps for full or complete trees.

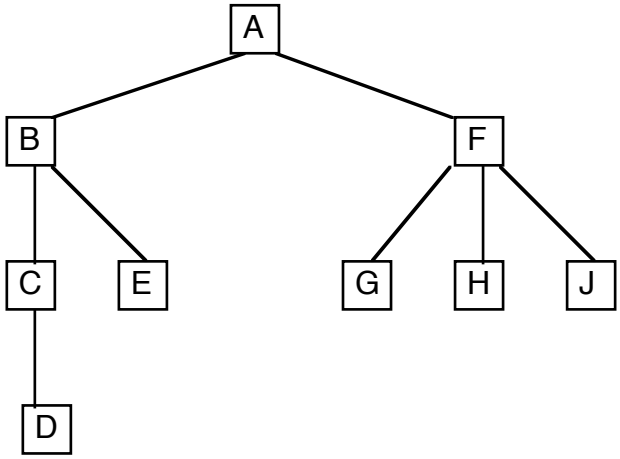
15. Represent the following ternary tree (degree 3 tree) in array notation, with no gaps for full or complete trees. For null tree elements draw a slash '/' in the corresponding array element. Write the array indices as well the contents of the array elements.



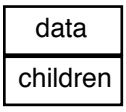


16. Describe how one converts a general tree into its corresponding binary tree.

17. Draw the diagram for the binary tree that corresponds to the following general tree .



18. Given the following node structure and field definition for a general tree. The data in each element of the positional sequence is a pointer to a child node in the tree.



int data
PositionalSequence children

Use **pseudocode** to complete the following function to determine the height of a subtree of a general tree where the root node of the subtree contains the given data value. You may use any data structures and their operations that we have discussed in class.

```
int heightSubtree(NODE root, int data)
```

```
        return height ( search ( root, data ) )
    end heightSubtree
```

A. Complete the algorithm for search. You may not assume an enumerator exists.

```
NODE search ( NODE tree, int data )
```

B. Complete the algorithm for height. You may not assume an enumerator exists. You must program the leaf check yourself.

```
int height ( NODE tree )
```

---

19. Describe the structure of a B+ tree. Draw a diagram showing a B+ tree of degree 2 where the tree:

a. has only two levels (the root node and leaf nodes); and
b. all the nodes are full.

Show the tree which results when one more data item is added to the tree.

---

20. Express formally (mathematically) the relationship between the depth of a binary tree and the number of nodes in the tree.

---

21. You are given a complete binary tree T represented by the array

```
T = array[1..15] of tree_node_type
```

where the tree nodes are labelled by the integers i, and $1 \leq i \leq 15$

List the node labels of T using:
a) pre-order traversal;
b) in-order traversal;
c post-order traversal.

---

22. Which of the following is true of a heap?

a) It is a binary tree.
b) It is a binary complete tree.
c) It is a search tree.
d) It is an AVL tree.

---

23. You are given a file called LIST containing N^2 distinct integers (no duplicates).

Assume the existence of the operations create_heap(h) and fix_heap(h,1,n) fix_heap is heapify in the course notes. Do not code the operations create_heap and fix_heap.

A. Using a heap data structure write pseudocode to print the N largest integers in descending order.

```
print(List list, int N) { ... }
```

B. What is the worst case computing time of your algorithm?

C. Illustrate with a diagram the contents of the heap after executing the algorithm on the example in part A above.

---

24. Assume that you have available an implementation of an abstract data type for binary search trees. The following operations are already implemented.

Require: node_one and node_two are both non null pointers to nodes in a binary search tree.
Ensure: The function compares the data portion of the tree nodes and returns equal if the data is equal in both tree nodes, less if the data in node_one^ is less than the data in node_two^, and greater if the data in node_one^ is greater than the data in node_two^.

```
compare(node_one, node_two : ^tree_node) : relation_type
```

Require: node is a non null pointer to a tree node.
Ensure: The data in the tree node is output on the terminal screen.

```
display(node : ^tree_node)
```

A. Write a recursive procedure to display the data parts of all pairs of parent-child nodes which violate the sorted ordering of the tree. Include either one of the diagnostic messages (a) `Child node should come after the parent', or (b) `Child node should come before the parent' with each violation.

```
verify(tree : ^tree_node)
```

---

25. Consider n integer values stored in the array K[1..n]. It is known that the values in K[1..n-1] (i.e. the first n-1 values in K) already form a heap. Write an efficient procedure that will make the entire array of n values a heap.

```
complete_heap(var K: array[1..K_max] of integer;n:integer)
```

---

26. The reverse level order traversal visits the nodes in a binary tree from left to right, level by level from the lowest leaf level up to the root. Write a procedure to display the data in the binary tree in reverse level order.
    You are to assume the following:
(i) A suitable implementation of the stack and queue abstract data types is available --- you do not have to code their operations.
(ii) Do not use recursion, use your own stacks and queues.

```
rev_traverse(tree: ^tree_node)
```

---

27. Given a diagram of a binary tree, list the nodes in:
(a) pre-order; (b) in-order; (c) post-order; (d) level order and left to right within level;
(e) reverse level order and right to left within level.

---

28. Give rules for the standard method of converting a general tree to a binary tree. Illustrate with an example.