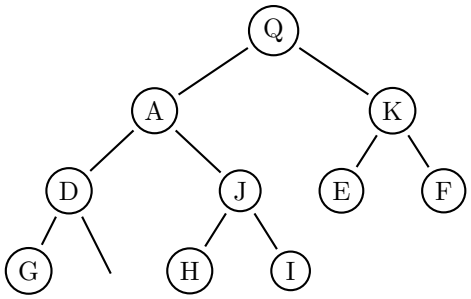
1. Traverse the binary tree using levelorder, preorder, inorder, and postorder:



2.
   - What is the maximum number of nodes in a tree with branching factor 2 and height h?

   - What is the maximum number of nodes on the last level in a tree with branching factor 2 and height h?

   - Approximately – what is the ratio of the number of nodes on the last level and the remaining nodes?

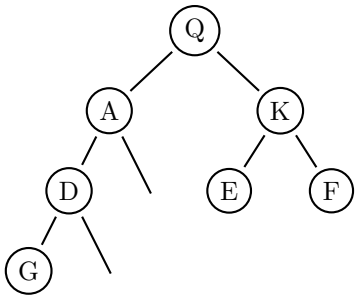
3. Answer the following questions using words (rather than code):

   - Explain how to find smallest element in a binary-search tree

   - Explain how to find successor of a given node in a binary-search tree

   - Explain how to delete a given node in a binary-search tree

4. Given the tree below



- Calculate balances in all nodes. You may add balances to the picture above.

- Rotate left around Q. Calculate balances in all nodes.

- Rotate right around Q. Calculate balances in all nodes.

```
a b c d e f g | h i j k l m n | o p q r s t u | v w x y z
0 1 2 3 4 5 6 | 0 1 2 3 4 5 6 | 0 1 2 3 4 5 6 | 0 1 2 3 4
```

5. Insert characters 'h', 'r', 'm', 'x', 'y', 'i' (in this order) into a hash using an array of size 7 and hashing function $h(ch) = (ch -' a')\%7$ and linear probing.

   Extra copies are in case you mess up:

0	
1	
2	
3	
4	
5	
6	

0	
1	
2	
3	
4	
5	
6	

0	
1	
2	
3	
4	
5	
6	

0	
1	
2	
3	
4	
5	
6	

0	
1	
2	
3	
4	
5	
6	

6. For each character in the hash calculate the number of probes required to find it.

h	
r	
m	
x	
y	
i	

7. For characters 'q', 'c', 'w' caclulate the number of probes required to declare the character is not in the hash.

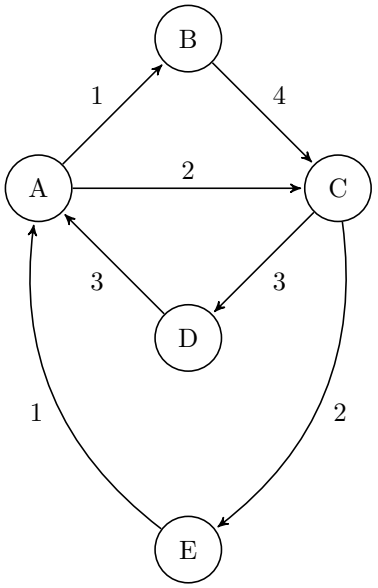
q	
c	
w	

8. (a) Build a **binary** min-heap (parent is smaller then children) from an array 5,3,9,2,1,7,4. **Show both tree and array** representations, show intermediate steps – the total number of tree-array pairs (steps) should be exactly the number of `heapify` calls.

   (b) Delete the top element, show intermediate steps.

9. (a) Show graphically a **binomial** min-heap with 7 elements (values 1,3,4,5,6,7,8).
   (b) Explain how a `union` operation will work on the previous heap and a heap with a single element 2.
   (c) delete the top element from the result of the union.

10. For the given directed graph



- fill in the table

node	in-degree	out-degree
A		
B		
C		
D		
E		

- write adjacency list representation of this graph

- write adjacency matrix representation of this graph

- what is the shortest cycle in this graph? _____
- what is the longest cycle in this graph? _____
- is this graph strongly connected? _____

11. Provide **iterative** implementation of breadth-first graph-traversal algorithm. Explain how to modify the algorithm to switch from BFS to DFS (depth-first).

12. For the following algorithm:

```
Alg1 ( int data[] ) { // data is provided as an unsorted array
    Collection coll = initialize collection from data;

    int m = maximum element of coll;
    for (int i = 0; i<100; ++i ) { //execute the loop 100 times
        delete m from the coll;
        insert m/2 into coll;
        m = maximum element of coll;
    }
}
```

Fill in the table below:

	initialize	maximum	delete	insert	total time to execute Alg1
binary heap					
unsorted array					
sorted array					
BST					

which data structure is the most efficient for this algorithm?

13. Execute Kruskal's algorithm on a given graph. Provide a list of edges in which algorithm processes them with a note whether it is added to MST or not.