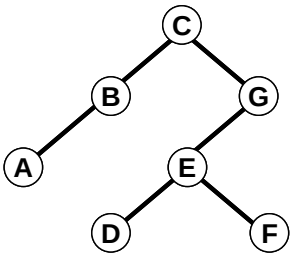
Breadth-first tree traversal scheme:
Step 0. Insert the tree root into the queue.
While quee is not empty do:
Step 1. Remove the first node from the queue and process it.
Step 2. Insert all children of the removed node into the queue.

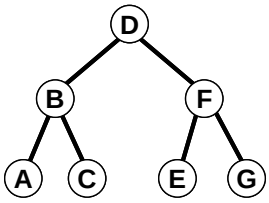
1. The tree depicted in the image is subjected to the BFS algorithm. At some point in time the contents of the queue is the following (we suppose that the queue head is on the left):

a) BGA
b) GA
c) AG
d) AEG
e) GEA
f) AE

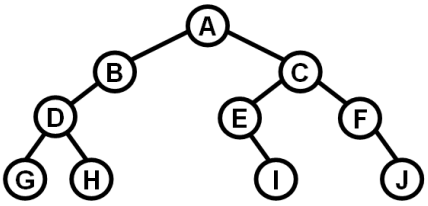


2. The tree depicted in the image is subjected to the BFS algorithm. At some point in time the contents of the queue is the following (we suppose that the queue head is on the left):

a) D
b) DF
c) FB
d) BGE
e) BAC
f) ABCD



3. Apply the BFS algorithm on the given tree and list the queue contents before the start of each Step 1 in the scheme above.



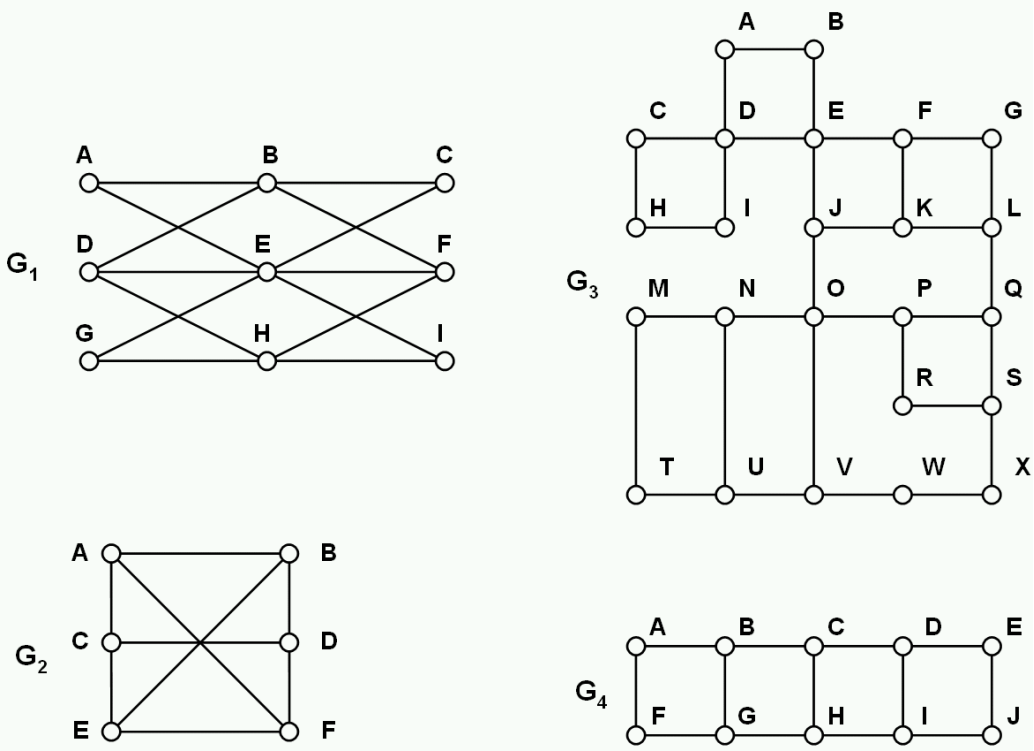
4. We have to reconstruct the shape the tree which was traversed by the BFS algorithm. There is the list of contents of the queue. Each element of the list (= queue contents) was printed immediately before the Step 1 in the BFS scheme above.

A
BC
C
DE
EFG
FG
GHI
HI
I

5. Write down a general algorithm which will reconstruct the tree and which input is the sequence of queue contents defined in the previous problem. The tree is supposed to be regular.

6. Consider the previous problem and suppose the following:
-- The processed tree to be reconstructed is a BST.
-- The queue stores the keys of the nodes.
Again, write down an algorithm which reconstructs this BST.

Graph examples:



In each of the following problems 7.-9., produce the list of visited nodes. The order of the nodes in the list should be the same as the order in which they were processed by the search/traversal algorithm.

7. Perform BFS on graphs G_1, G_2, G_3, G_4. Each time start in node A. In each node of each graph always process the neighbouring nodes in alphabetical order.
For example, in node B in G_2 process the neighbours in order A, D, E.

8. Perform DFS (Depth-first search) on graphs G_1, G_2, G_3, G_4. Each time start in node B. In each node of each graph always process the neighbouring nodes in alphabetical order.

9. Perform DFS (Depth-first search) on graphs G_1, G_2, G_3, G_4. Each time start in any node you like. The order in which the neighbours are processed is arbitrary. The additional condition is that each two neigbouring nodes in the list of visited nodes should also be neighbours in the graph. In other words, the search tree produced by DFS should contain no branchings and it should be one simple path.