



# HABIB UNIVERSITY

## Data Structures & Algorithms

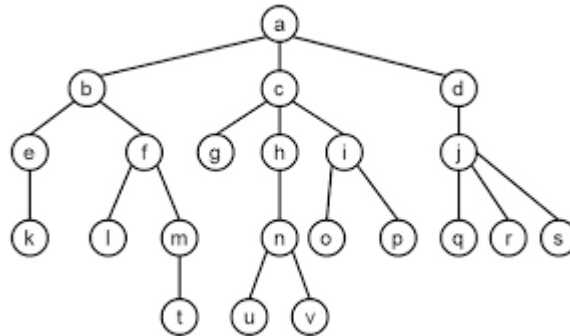
CS/CE 102/171 Spring 2023

Instructor: Maria Samad

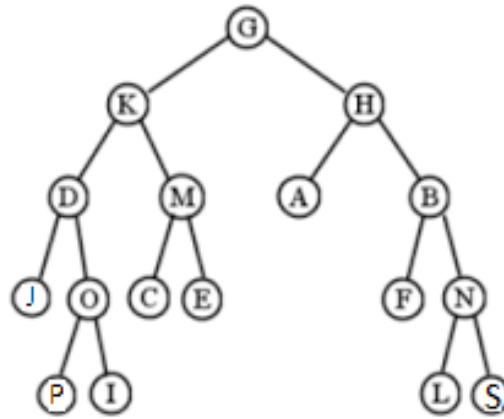
### Trees Functions

Student Name: \_\_\_\_\_

For the given trees, what will be the output of the Tree functions as specified? If a particular question is not applicable to the given tree, then simple state N/A in your answer

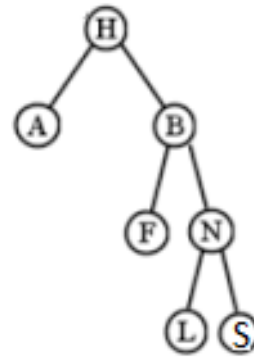


- degree(h): **2**
- degree(u): **1**
- indegree(i): **1**
- indegree(o): **1**
- outdegree(j): **3**
- outdegree(q): **0**
- T.root(): **Node a**
- T.is\_root(c): **False**
- T.is\_root(a): **True**
- T.parent(f): **Node b**
- T.num\_children(j): **3**
- T.children(c): **[Node g, Node h, Node i]**
- T.num\_children(p): **0**
- T.children(k): **[ ]**
- T.is\_leaf(l): **True**
- T.is\_leaf(n): **False**
- len(T): **22**
- T.is\_empty(): **False**
- iter(T): **[Node a, Node b, Node c, Node d, Node e, Node f, Node g, Node h, Node i, Node j, Node k, Node l, Node m, Node n, Node o, Node p, Node q, Node r, Node s, Node t, Node u, Node v]**
- depth(T, m): **3**
- depth(T, v): **4**
- height(T, a): **5**
- height(T, g): **1**
- level(b): **1**
- level(i): **2**
- Path from Node a to Node r: **Node a → Node d → Node j → Node r**
- Path from Node d to Node e: **Not possible**
- T.left(c): **N/A**
- T.right(h): **N/A**
- T.sibling(g): **N/A**
- Is this a Binary Tree or General Tree? **General Tree**
- Is this a Proper or Improper Binary Tree? **N/A**
- Draw the right subtree of Node e:  
**N/A**
- Calculate Balance Factor (show working):  
**N/A**
- Is it a balanced binary tree? Give reason for your answer:  
**N/A**
- Is it Left-Heavy or Right-Heavy or a Perfectly Balanced Binary Tree? **N/A**

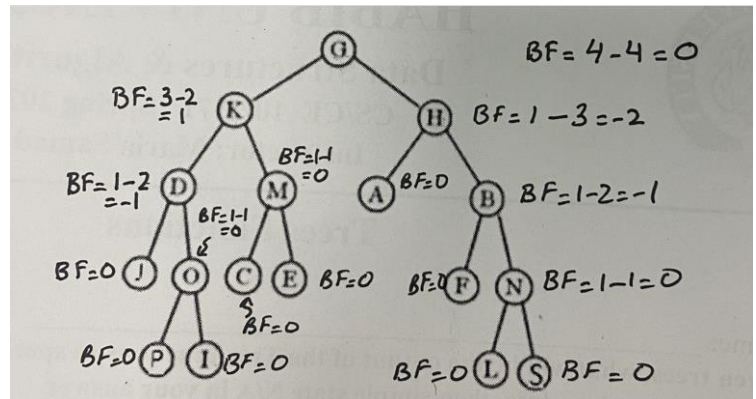


- degree(**B**): **3**
- degree(**I**): **1**
- indegree(**A**): **1**
- indegree(**G**): **0**
- outdegree(**D**): **2**
- outdegree(**P**): **0**
- T.root( ): **Node G**
- T.is\_root(**C**): **False**
- T.is\_root(**G**): **True**
- T.parent(**E**): **Node M**
- T.num\_children(**F**): **0**
- T.children(**N**): [**Node L, Node S**]
- T.num\_children(**H**): **2**
- T.children(**J**): [**]**
- T.is\_leaf(**A**): **True**
- T.is\_leaf(**L**): **True**
- len(T): **17**
- T.is\_empty(): **False**
- iter(T): [**Node G, Node K, Node H, Node D, Node M, Node A, Node B, Node J, Node O, Node C, Node E, Node F, Node N, Node P, Node I, Node L, Node S**]
- depth(T, **K**): **1**
- depth(T, **S**): **4**
- height(T, **O**): **2**
- height(T, **G**): **5**
- level(**M**): **2**
- level(**G**): **0**
- Path from **Node G** to **Node I**: **Node G → Node K → Node D → Node O → Node I**
- Path from **Node H** to **Node L**: **Node H → Node B → Node N → Node L**
- T.left(**D**): **Node J**
- T.right(**F**): **N/A**
- T.sibling(**L**): **Node S**
- Is this a Binary Tree or General Tree? **Binary Tree**
- Is this a Proper or Improper Binary Tree? **Proper Binary Tree**

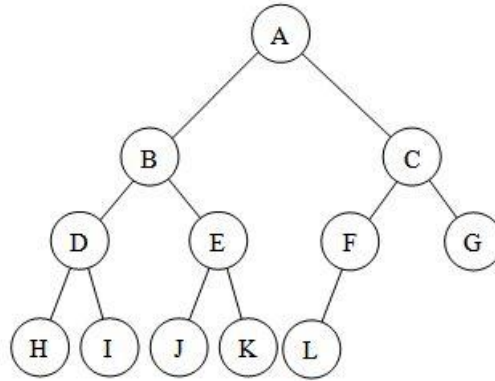
- Draw the right subtree of **Root Node**:



- Calculate Balance Factor (show working):



- Is it a balanced binary tree? Give reason for your answer: **No, because Node H gives a balance factor of more than 1, i.e. 2**
- Is it Left-Heavy or Right-Heavy or a Perfectly Balanced Binary Tree? **Neither because it is not balanced to begin with**

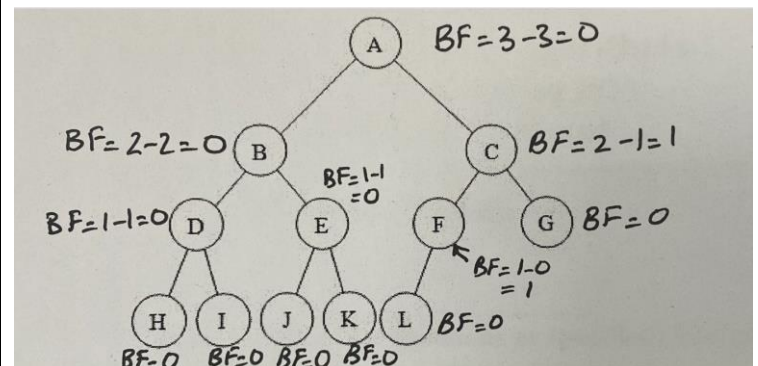


- degree(D): **3**
- degree(A): **2**
- indegree(B): **1**
- indegree(C): **1**
- outdegree(E): **2**
- outdegree(G): **0**
- T.root(): **Node A**
- T.is\_root(F): **False**
- T.is\_root(A): **True**
- T.parent(H): **Node D**
- T.num\_children(I): **0**
- T.children(J): **[ ]**
- T.num\_children(A): **2**
- T.children(B): **[Node D, Node E]**
- T.is\_leaf(H): **True**
- T.is\_leaf(A): **False**
- len(T): **12**
- T.is\_empty(): **False**
- iter(T): **[Node A, Node B, Node C, Node D, Node E, Node F, Node G, Node H, Node I, Node J, Node K, Node L]**
- depth(T, A): **0**
- depth(T, D): **2**
- height(T, A): **4**
- height(T, B): **3**
- level(B): **1**
- level(E): **2**
- Path from Node A to Node L: **Node A → Node C → Node F → Node L**
- Path from Node K to Node A: **Not Possible**
- T.left(C): **Node F**
- T.right(H): **None**
- T.sibling(L): **None**

- Draw the right subtree of Node E:



- Is this a Binary Tree or General Tree? **Binary Tree**
- Is this a Proper or Improper Binary Tree? **Improper Binary Tree**
- Calculate Balance Factor (show working):



- Is it a balanced binary tree? **Yes, because all the nodes have balance factor of not more than 1**
- Is it Left-Heavy or Right-Heavy or a Perfectly Balanced Binary Tree? **Balanced Binary Tree at the Root Node is perfectly balanced**