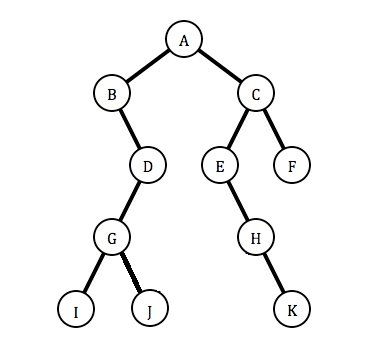
# Instructions

* You may work with a partner for this assignment. If you do, you may only work with **one** partner. Hand in the assignment only once to the dropbox, but clearly indicate who both members of the pair are.
* Complete Part A in a Word document.
* Complete Part B in a copy of your code from class.
* Put your Word document in the root of your code project. Submit the entire folder to the folder on Dropbox. Make sure you include the Lists, and BinarySearchTrees projects – everything needed to run your code should be in there.
* Due date: 9th October, 2024.

# Part A Written [20 marks]

1. Consider the following binary tree. [3 marks] **DONE**
   1. Write the preorder, inorder and postorder traversals of the binary tree shown below.  
      

Preorder: A, B, D, G, I, J, C, E, H, K, F

Inorder: I, J, G, D, B, A, K, H, E, C, F

Postorder: I, J, G, D, B, K, H, E, F, C, A

1. A binary tree(not necessarily a *search* tree – no guarantee that the nodes are in any particular order) has a preorder traversal of Fred-Nancy-Kyle-Jack-Bob-Sam-Greg-Ted and an inorder traversal of Kyle-Nancy-Jack-Fred-Sam-Ted-Greg-Bob. What is its postorder traversal? How did you work this out? [3 marks] **DONE**

Kyle, Jack, Nancy, Ted, Sam, Greg, Bob ,Fred

Fred was the root node, so that had to be at the start

Kyle Nancy and Jack had to be on the left and Sam Ted Greg and Bob were on the right

Post order is LRN so the left would be Kyle Jack Nancy

The right would be Ted Sam Greg Bob

Add Fred to the end of it

1. If a BST stores 1 billion nodes, [4 marks] **DONE**
   1. What are the minimum and maximum height of the tree?
      1. Min = floor (log2(n)+1); min = 30
      2. Max = n-1; max = 999999999
   2. If the tree were an AVL tree, what would be the approximate maximum number of levels?
      1. Max of AVL = 1.44 \* log2n; Max of AVL = 43.056 levels

(\*\***show any formulas used, and cite your references for formulas you had to look up**\*\*)

Min = floor (log2(n)+1)

Max = n-1

Max of AVL = 1.44 \* log2n

<https://www.geeksforgeeks.org/relationship-number-nodes-height-binary-tree/>

<https://www.geeksforgeeks.org/practice-questions-height-balancedavl-tree/>

<https://www.omnicalculator.com/math/log-2>

1. For the following tree, show the resultant tree assuming the Remove method from class was used. For each question below, assume you are removing from the original tree. Show all intermediate steps and explain what is going on in each step. [6 marks] **DONE**
   1. Remove node 76 from the original tree.
      1. Make the node of 76 null
      2. Shift all the child nodes up one spot
      3. So the right subtree of 50 is now 54, 72, 67
   2. Remove node 23 from the original tree.
      1. Copy all values of child nodes of 23 and make 23 null
      2. Shift them up a spot
      3. The right subtree of 17 is now 19, 21
   3. Remove node 50 from the original tree.
      1. Find the successor of 50 (smallest in the right tree = 54)
      2. Copy the content of the successor (54) to the node (50 now = 54)
      3. Move all the child nodes up (72 is now a child node of 76, 67 is still a child of 72)

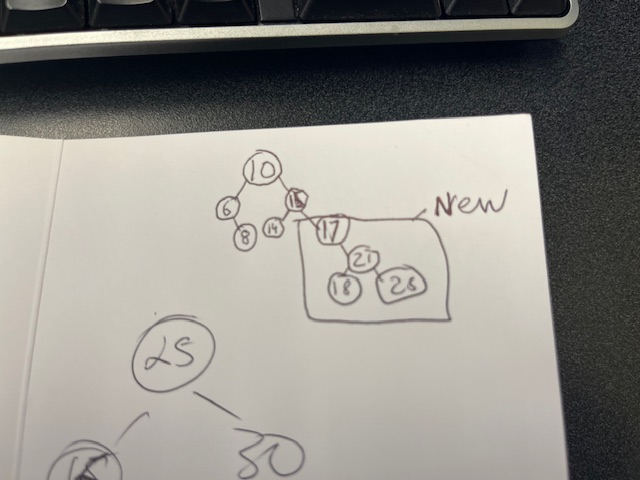
A picture containing clipart

Description automatically generated

1. Given the following AVL tree that balances when adding, show the rotations that would take place after adding 18 to the tree. Show the intermediate steps as well as the final balanced tree [4 marks] **DONE**

* + 1. 18 is greater than 10 and 15 so it is on the right subtree of 15
    2. It is less than 21 so its on the left tree of 21
    3. Is it greater than 17 so it is the right leaf node of 17, however its unbalanced
    4. Perform a left rotation, makin the right child of 15 17. Then the right subtree of 17 is 21. With a left subtree of 21 being 18 and a right tree being 25

A diagram of a tree

Description automatically generated

# Part B Code [30 marks]

All code should follow the proper coding standards and be well-written, relatively efficient, and well-commented. Poorly written or hard-to-read code may lose marks even if it solves the problem. Also ensure you include driver codes in your Program.cs file

1. Implement the following two methods of the LinkedList done in the class: [ 6 marks]
   1. InsertLast(int index), that inserts a new node with data “index” at the end of the linked list. **DONE**
   2. DeleteLast(), that deletes the last node of the linked list. **DONE**
2. Write code for a method of your BST class that will create a sorted (descending order) linked list from the provided tree. The linked list from the project built in class will be used. If the root is null, an empty list will be returned. [9 marks] **DONE**

Use the given BST in your driver code:

A black background with a black square

Description automatically generated with medium confidence

1. The approach used to code the Remove method of the Binary search tree done in class was to find the successor (smallest in the right subtree) of a given node if it has a right child, and then replace the node with the successor. Modify this code by writing a RemovePred method that finds the predecessor (largest in the left) of the node to be removed, and replaces it accordingly. [ 5 marks] **DONE**
2. Include a RemoveBalance method in Binary Search/ AVL Tree done in class that removes a node and balances it as appropriate. [ 5 marks] **DONE**