**Test 1**

Let Answers are written in this font

1. Prove that the depth of a binary search tree is O(log N) on average.

* The depth of the tree is equal to the height of the root.
* Assuming that the tree is a perfect binary tree, each node has two children. This gives us the total number of nodes in a tree as the sum of the nodes at all levels: node *N*.

When we calculate the total number of nodes in a tree, we get the summation formula as:

The max number of nodes N:

Therefore, depth of the tree =

1. Design a linear-time algorithm that verifies the height information in an AVL tree is correctly maintained and the balance property is in order.

Assuming every node in an AVL tree has the fields

* 1. Data
  2. Height
  3. Left
  4. Right

1. Create a function called AVLVerifier() which returns a Boolean
2. This function visits every node in the tree
3. If the node is present then:
   1. Check the left and right node for the height
      1. Verifies that the height differs by 1
      2. If this is true, go to step III
      3. Otherwise function returns false

As the function recursively visits every node, every step takes N number of times, which constitutes to run time of O(N), which shows that the algorithm is executed in linear time.

1. Create a binary expression tree for a postfix expression.

Using Postfix expression: abc\*+de\*f+g\*+

Processing every character for expression:

1. Push a onto the stack
2. Push b onto the stack
3. Push c onto the stack
4. Pop c and b and make them children of the node \* and push subtree onto the stack (T­­1)
5. Pop T1 and node a and make them children of new subtree of + node and then push the new subtree onto the stack (T2)
6. Push d onto the stack
7. Push e onto the stack
8. Pop e and d and make them children of new subtree of \* node and then push the new subtree onto the stack (T3)
9. Push f onto the stack
10. Pop T3 ad f and make them children of new subtree of + and then push the new subtree onto the stack (T4)

Figure

1. Write a program to evaluate a postfix expression and convert a postfix expression to infix expression.
2. A binary search tree with N nodes has N+1 null references, half the space allocated in a binary search tree for links is wasted. Suppose that if a node has a null left child, we make its child link to its inorder predecessor and if a node has a null right child, we make its right child link to its inorder successor, making a threaded tree.
3. How to distinguish threads from real children links?
4. Write routines to perform insertion and deletion into a tree threaded in this manner.