Symbol Tables:→

Searching is also one of the fundamental operations in computer science. Linear search and binary search are two most important and widely used techniques.

In both of the algorithms the location of an item is determined by a sequence of comparisons.

In each case, the item is to be searched is repeatedly compared with items in certain locations of the list.

For a collection of n items, linear search requires O(n) comparisons whereas for binary search, $O(\log_2 n)$ comparisons are required. We know binary search is an efficient technique than linear search. But in some situations these algorithms is also too slowly.

To speed up the searching a symbol table can be used.

Symbol table constructed by compiler stores identifiers and information about them. The speed with which this table can be constructed and searched is critical to the speed of compilation.

Static tree table and dynamic tree table are the just new names and the concepts behinds the naming these so is already known to you.

As we know as a program is being translated the compiler creates a table called a symbol table. When a variable is first defined, it is entered into the symbol table, and a memory location is assigned to it.

The association of memory address with a variable name is called binding. The time when this binding ~~occur~~ performed is called binding time. This time may occur either compile time or execution time depends on the types of variables used.

programming languages are usually classified as having either static allocation or dynamic storage allocation.

Static storage allocation

associates variables with memory location at compile time; whereas dynamic storage allocation associates variables with memory locations at execution time.

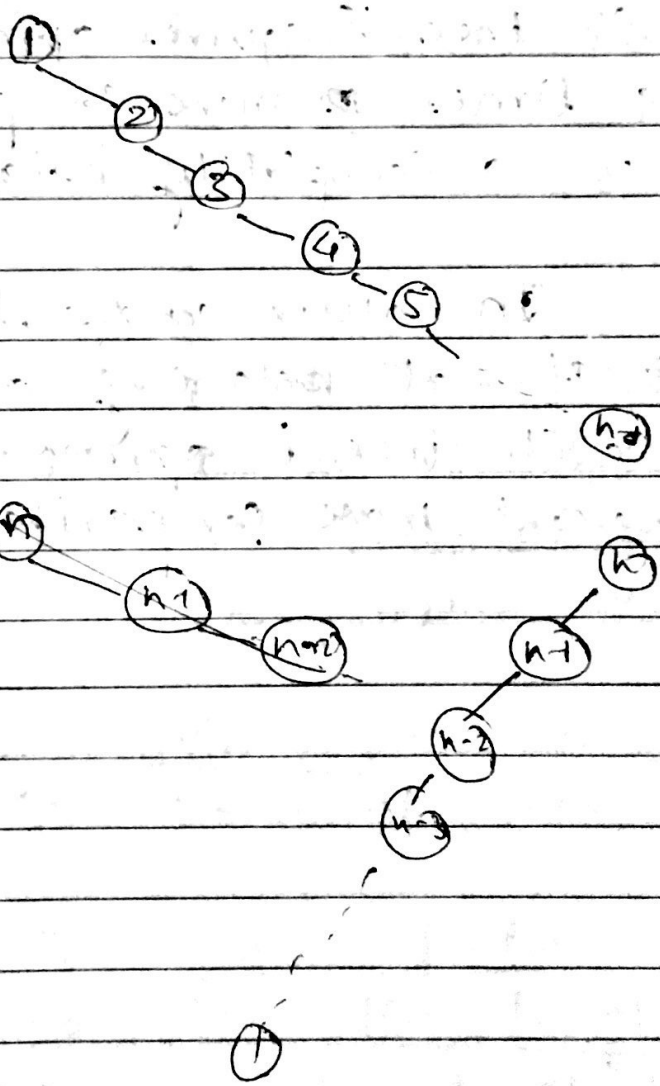
A language which is implemented with static storage allocation does not support Recursion.

Because in static storage allocation each variable is alloted a single memory location at compile time. But a variable needs multiple versions of it while generating Recursive calls. Because in Recursion intermediate results are to be stored in multiple versions. if not so then previous values may be overwritten or lost.

Thus the table which is created during compile time is called Static tree table and table which is created during execution time is called dynamic tree table.

Dynamic Tree Tables :→

In binary Search Tree suppose we have to insert n nodes and in such a case suppose first node is the root and all (n-1) nodes are as left subtree of right subtree, or in worst case it may be left skewed or right skewed tree.



So if as levels of tree increase the number of comparisons also increased.

To Reduce the no. of comparisons we balance the binary tree as possible.

Thus we balance the binary tree as much as possible then no. of comparisons can be greatly reduced.

Because the average binary search tree requires approximately 1.39 times as more comparisons than a completely balanced tree.

In other words, the average cost of not balancing a binary search tree is approximately 39 percent more comparisons.