

Linked list:-

\* This implementation ~~is~~ used linked list. A link field is added to each record

\* Searching of names is done in order pointed by link field

\* A pointer "first" is maintained to point to first record of symbol table

\* Insertion is fast  $O(1)$ , but look up is slow for large table -  $O(n)$  on average.

Binary search Tree:-

\* Another approach to implement Symbol table is to use binary search tree i.e. we add two link fields i.e. left and right child.



- \* All names are created as child of root node that always follows the property of binary search tree
- \* Insertion and lookup are  $O(\log_2 n)$  on average

## Hash Table

- \* In hashing scheme two tables are maintained a hash table and symbol table and is the most commonly used method to implement these symbol tables
- \* A hash table is an array with index range 0 to table size - 1. These entries are pointers pointing to names of symbol table
- \* To search for a name we use hash function that will result in any integer between 0 to table size - 1
- \* Insertion and lookup can be made very fast -  $O(1)$
- \* Advantage is quick search is possible and disadvantage is that hashing is complicated to implement.