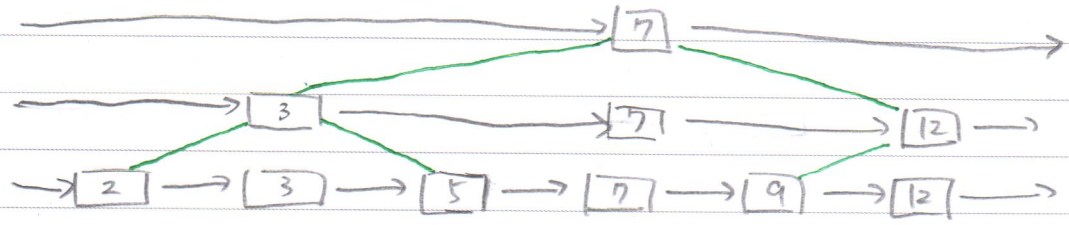
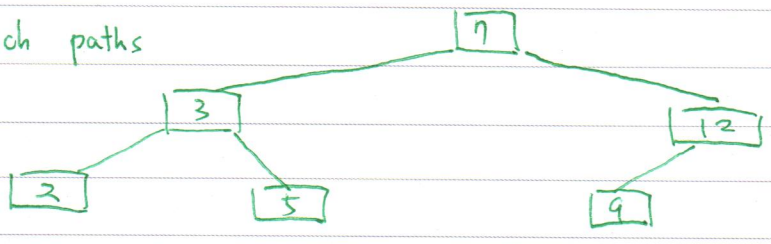
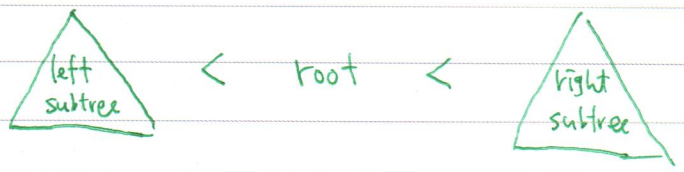
* skip list to BST



the search paths





for every subtree : called binary search tree

* search for key on BST

```
BST-Search (k, T) {
    mid = root of T
    if (k < mid)
        return BST-Search (k, T.left);
    if (k > mid)
        return BST-Search (k, T.right);
    else
        return mid.value;
}
```

worst case : O(h) w/ h being height

* insert : similar to search

* remove :

leaf : simple

one child : simple

two children : find right-most decendent of left-subtree O(h)