

**Consider following implementation of BST for this lab:**

**Task 1:** Implement class Linked list with tail pointer. Take integer, data type in Node. Create following functions:

- addAtTail
- search
- print (to print complete linked list)

**Task 2:** Take an array of 40 elements. Initialize with random values (between 10 & 99); take care that no value should repeat. Now take array of Linked List (Implemented in task 1) of size 20. Use following hash function, and add 20 elements of array in respective linked list according to hash value. Finally print all 20 linked lists (each in a row), so that once can observe how many elements are stored in each linked list ?

**Task 3:** Use given code of Max-Heap. Simulate priority queue of size 20 at max, call add & remove function at random, print each value added in priority queue & print each value removed from priority queue. Show minimum 20 operators include add & remove?

```
int left(int i){
    return (i+1)*2-1;
}
int right(int i){
    return (i+1)*2;
}
int parent(int i){
    return (i-1)/2;
}
void swap(int *a, int i, int j){
    int temp=a[i];
    a[i]=a[j];
    a[j]=temp;
}
void increaseKey (int *data, int i, int val){
    while (i!=0 && data[parent(i)] < data [i]){
        swap(data, i, parent(i));
        i = parent (i);
    }
}
void decreaseKey (int *data, int i, int size){
    while (left(i)< size){
        int largest=left(i);
        if (right(i)<size && data[right(i)]>data[left(i)])
            largest= right(i);
        if (data[i]<data[largest]){
            swap(data, i, largest);
            i = largest;
        }
        else return;
    }
}
```

\*\*\*\*\* END \*\*\*\*\*