Data Structures C & D
FAST-NU, Lahore, Fall 2016

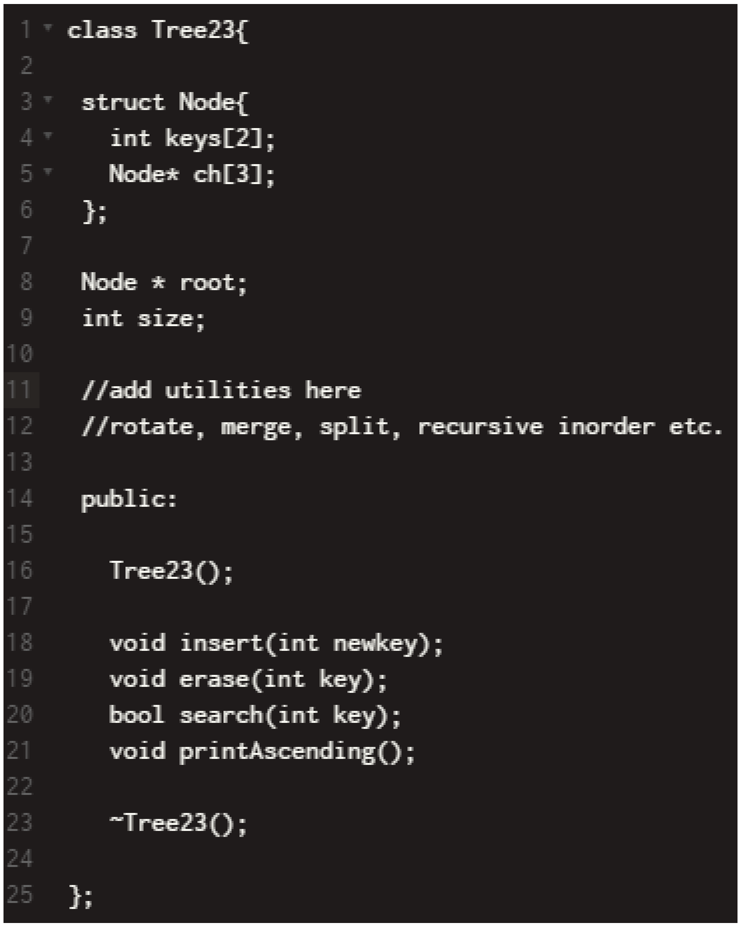
Homework 7

Height Balanced Search Structures

Due: Friday, November 4, 11:55PM — Marked out of 50 points.

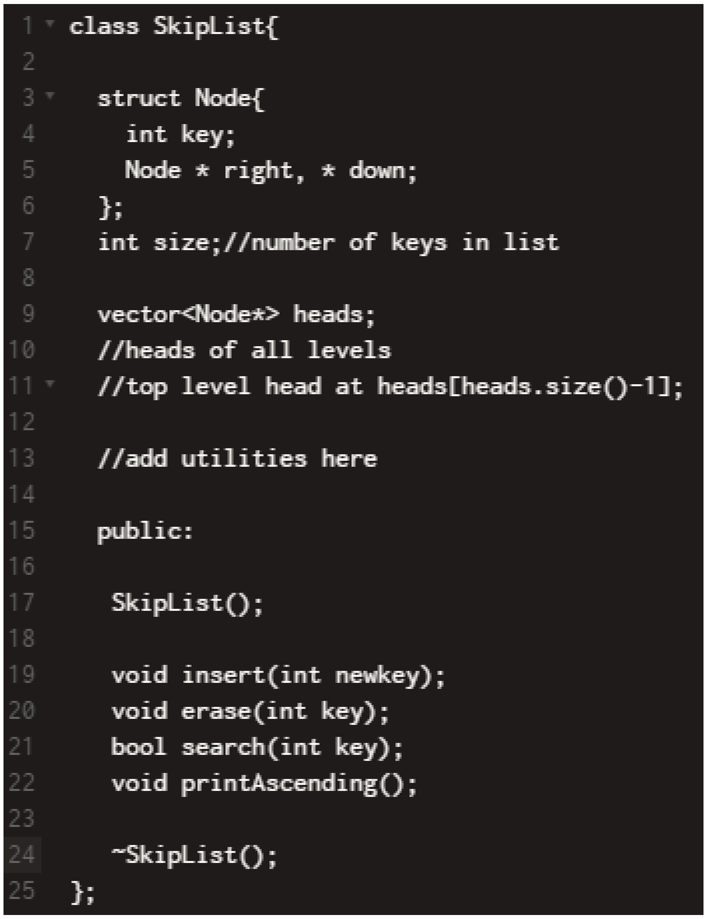
This is a straight forward assignment. You have to implement either a 2-3 Tree or a Skip List (one of them) to store integers (always integers, template code not required). Your program should provide a simple menu on the console to insert, delete and search integers in the structure; there should be a fourth option to print all keys in ascending order. Your data structure should only accept unique keys. Following are the starting points for the code (you should add to them, and must follow the essential algorithms explained in class).

1. **Starting code skeleton for class Tree23**



```
class Tree23{

 struct Node{
   int keys[2];
   Node* ch[3];
 };

 Node * root;
 int size;

 //add utilities here
 //rotate, merge, split, recursive inorder etc.

 public:

   Tree23();

   void insert(int newkey);
   void erase(int key);
   bool search(int key);
   void printAscending();

   ~Tree23();

};
```

2. Starting code skeleton for class SkipList



```
class SkipList{

  struct Node{
    int key;
    Node * right, * down;
  };
  int size;//number of keys in list

  vector<Node*> heads;
  //heads of all levels
  //top level head at heads[heads.size()-1];

  //add utilities here

  public:

   SkipList();

   void insert(int newkey);
   void erase(int key);
   bool search(int key);
   void printAscending();

   ~SkipList();
};
```

* THE END *