CS201c: Practice Lab 2
Instructor: Apurva Mudgal
Friday, 16th August 2019

1 Binary search trees using C++ templates

1. Read about how to use templates in C++.
2. Complete the following C++ templates for binary search tree:

```
template <typename T>
class Node {
        private:
                T x;
                Node<T> *left; // left child
                Node<T> *right; // right child
                Node<T> *parent; // parent node
                // any other augmented information

        public:
                //define suitable functions here
};


template <typename T>
class BST {
        private:
                Node<T> *root; // root node
                int n; // total number of nodes

        public:
                // define suitable constructor, destructors, etc. here.

                int search(T x); // search x in BST
                int insert(T x); // insert x in BST
                int remove(T x); // delete x from BST

                // return k-th smallest data in the tree
                T order_statistics(int k);
```

```
};
```

Here the operators $<, >, ==, <=, >=, !=$ are overloaded for type T.

Define instances of above class templates for different data types T, and test that they work correctly.

2 Performance of binary search trees on randomly ordered input

Without loss of generality, assume that the keys to be inserted in a BST are $1, 2, \ldots, n$. Let $(\sigma(1), \sigma(2), \ldots, \sigma(n))$ be a *random permutation* of $(1, 2, \ldots, n)$ (i.e., each of the $n!$ permutations are equally likely to be σ).

Suppose we insert keys $\sigma(1), \sigma(2), \ldots, \sigma(n)$ in an empty binary search tree in this order. Let T_σ be the resulting binary search tree.

Your objective is to experimentally estimate the average height of tree T_σ. To be specific, let $h(T_\sigma)$ be the height of tree T_σ. Then, average height for a random permuation is $\frac{\sum_\sigma h(T_\sigma)}{n!}$.

Note. (i) You can try $n = 128, 256, \ldots, 65536$ (successive powers of 2). For each n, you may generate $K = 10000$ permutations. Let height of binary search trees for these permutations are $h_1, h_2, \ldots, h_K$. Then, average height (of a random binary search tree) for n keys can be estimated by $\frac{h_1+h_2+\ldots+h_K}{K}$.

Plot this average height as a function of n. What do you observe?

Question: Can you conclude that if the elements to be inserted in a BST are given beforehand, a good strategy is to randomly permute them before constructing the BST?