CpSc 2120: Algorithms and Data Structures

Instructor: Dr. Brian Dean — Fall 2020
Webpage: `http://www.cs.clemson.edu/~bcdean/` — MWF 9:05-9:55
Handout 6: Lab #4 — Flour 132

1 Binary Search Tree Practice

The goal of this lab is to gain familiarity with binary search trees. You will code a number of primitive operations on a BST, ultimately building the code base for a "randomly-balanced" BST. As will be described in lecture, this form of balancing ensures that the BST stays in a state which is always as if it was just built from scratch by inserting its elements in random order, so its height is always $O(\log n)$ with high probability, and every major BST operation therefore runs in $O(\log n)$ time with high probability.

You will start with the following file:

`/group/course/cpsc212/f20/lab04/bst.cpp`

In this file you will find the structure defining a node in a BST, which contains an integer key, pointers to its left and right children, and a "size" field that should contain a count of the number of nodes in its subtree. For convenience, there is also a constructor that lets you easily allocate a new node given its integer key; for example:

```
Node *n = new Node(7);
```

There are a number of functions to fill in for this lab, all marked with a "TBD" comment. All of them start with a comment describing what the function should do (in fact, we have already written code for some of them during class, so for example `insert` and `print_inorder` should be quite easy as a warm-up). A few notes on the more advanced functions:

- You should `remove` a node from a tree by replacing it by the join of its two subtrees. Don't forget to de-allocate the memory used by the node you remove.
- When you `join` two subtrees L and R together, you'll make a random choice between the root of L and the root of R for the root of the merged tree (this helps with balancing). You want to choose with probabilities proportional to the sizes of L and R, so you will choose the root of L with probability $\frac{|L|}{|L|+|R|}$, or the root of R with probability $\frac{|R|}{|L|+|R|}$ (here $|T|$ denotes the size of T). To help with making your random choice: the function `rand()` (defined in `cstdio`) returns a random integer, so the expression `rand() % N` returns a random integer in the range $0 \ldots N-1$. For example, "`if (rand() % N == 0) A; else B;`" would execute statement A with probability $1/N$ (only if the random number we choose comes out to zero), and otherwise statement B with probability $1 - 1/N$. Code for making this random choice is provided for you already, but you should make sure you understand what it is doing.

- The `split` function needs to return pointers to two trees, and since functions can generally only return one value, we have therefore achieved this by returning a "pair" object consisting of two Node pointers. A pair of two things a and b is created with the `make_pair(a,b)` function, as illustrated in the lab code. If `P` is a pair, then `P.first` and `P.second` are the two objects that form the pair. We will find C++ pairs to be quite useful in several situations moving forward. To use pairs or their more sophisticated relatives, tuples, you'll need to compile your code with C++11 extensions enabled, using the compiler flag `-std=c++11`.

For convenience, the `main` function contains some pre-built testing code, which can help you determine if your implementations are working correctly. You may want to put a temporary "return" statement in the middle of this code while working on the lab so that you only execute the testing code for the functions you have built thus far (as the rest of the testing code might crash when trying to run functions you haven't yet implemented).

2 Grading

For this lab, you will receive 8 points for correctness and 2 points for having well-organized, readable code. Zero points will be awarded for code that does not compile, so make sure your code compiles on the lab machines before submitting!

Final submissions are due by 11:59pm on the evening of Monday, September 28. No late submissions will be accepted.