Assignment: Balanced Binary Search Tree

CS3334 Teaching Team

City University of Hong Kong

November 3, 2021

Learning Outcomes

- Enhance coding skills through practice;
- Implement the most popular types of balanced binary search trees (BBSTs);
- Evaluate and compare similar BBSTs and find their strengths and weaknesses;
- Learn how to select appropriate data structures to solve problems.

1 Overview

You are required to maintain a multiset S, which supports the following operations:

1. Insert a number x to S;
2. Erase a number equal to x from S if exists;
3. Find the order of x in S, i.e., $1 + \sum_{v \in S} [v < x]$;
4. Find the x-th smallest number in S;
5. Find the precursor of a number x in S, i.e., $\max\limits_{v \in S, v < x} \{v\}$ if exists;
6. Find the successor of a number x in S, i.e., $\min\limits_{v \in S, v > x} \{v\}$ if exists.

2 Detailed Requirements

2.1 Select BBSTs

In this assignment, you are asked to select and achieve any **two** kinds of BBSTs mentioned or introduced in the lecture (preferably Splay and AVL Tree) to achieve the operations listed above.

You need to submit your implementations to the Online Judge System and pass the corresponding question BBST.

2.2 Efficiency Analysis and Study Report

You will be the problem setter this time to design the test data for this problem. You are required to **compare and analyse the differences in efficiency** between different BBSTs by feeding them **different types of testing data** designed and generated on your own for the operations mentioned above. The generator programs should be written in C++, and the data you generated should meet the constraints in the problem description. A sample generator is available on canvas.

Then, you need to write a report to brief your progress and findings on this assignment. The following things should be clearly stated in the report:

- The compiling command for compiling your generator programs, e.g., `g++ sampleGenerator.cpp -o exec_file`. Some available compiling options are `-std=c++11, -lm, -O2`;
- An explanation about the strategy and purpose of the test data sets you designed and generated;
- An analysis of the performance of the different BBSTs under your different data sets;
- A conclusion including the strengths and weaknesses of the BBSTs you implemented based on the performance analysis.

3 Assessment Criterion

You will be assessed by how much and how well you can apply what you have learnt from the course, some considerations are:

- Whether you successfully implemented exactly two kinds of the BBSTs and your programs passed the problem in the Online Judge System;
- To what extent the data generated by your generator programs matches the design stated in your report;
- To what extent the data sets you designed and stated in your report can **reveal the characteristics of the BBSTs**. For example, if you implemented a Splay tree, you may think about under which kind of special data it will perform well;
- How comprehensively you compared the performance of different BBSTs under your test data;
- To what extent your conclusion matches the performance of your implementations under the test data and how comprehensively you analysed the strengths and weaknesses of your BBSTs.

4 Hints

- One reasonable standard for assessing the performance of the BBSTs is the number of rotations during the whole process.
- You may start with a further research about the BBSTs by reading some references.

5 Due Date and Submission

- The due date is Dec 2; the deadline time is 11:55 pm.
- Submit your codes (implementations of two different BBSTs) to the Online Judge System directly.
- Submit a zip file containing all your materials, i.e., codes, generator programs, study report. The codes submitted to the Online Judge System should be included in this file as well. **You are not supposed to include data set files in your submission, so please make sure the generator programs work.**