

Programming Project 2

Due **11/3/11** before class

Red-Black Tree Construction (20 points)

Given a sorted array of n numbers, the task is to construct a red-black tree in $O(n)$ time. (This means you cannot repeatedly insert one number after the other, since this would yield a runtime of $O(n \log n)$, and zero points on this project.)

Here is an idea on how to accomplish this task:

1. First, construct an almost-balanced binary tree as follows: Pick the median (by array lookup), create a vertex for it, and assign the left half of the array to the left child and the right half of the array to the right child. Then repeat (or recurse) similarly on both children.
2. Once you have created the tree, figure out how to assign colors to the vertices in order to ensure it is a valid red-black tree.

Your code and project directory should contain the following:

- A way to insert a sorted array of numbers. (Hard-coding is fine if documented properly.)
- A way to output the resulting red-black tree in a visual way. Important node information to display are key, color and black-height. (You can use or modify code available from the web for this if you like.)
- A short test-case report that documents some test cases that you test your code with.

Turnin instructions

- You are allowed to turn in this programming project in groups of two.
- You can use Java, C, or C++ for this project. If you want to use a different programming language, check with our TA first.
- **The name of your project directory should be** `project2_<language><lastName1><firstName1><lastName2><firstName2>`, where `<language>` is the programming language you are using.
- Zip up a directory with your entire project (source code and test case report). Turn in the zip file by uploading it to Blackboard. In the comments section during the upload to Blackboard please add instructions on how to compile the program and on how to run the test cases.
- All projects need to compile. If your program does not compile you will receive 0 points on this project.
- Do not use any fancy libraries. We should be able to compile it under standard installs of Java, C, or C++ under linux and/or windows.