CSc 130 Assignment 2
Simulation of A Simple CPU Scheduler Using AVL Tree

Dr. Jinsong Ouyang

1 Requirements

1. Design and implement an AVL Tree Node class that must support “findMin”, “insert” and “remove” operations.
2. Design and implement a driver (the main method) that does the following:
   (a) Creates an array that contains a list of 5000 integers from 0 to 4999 sequentially.

   (b) Randomly shuffle the array so that the values in the array are not in order.

   (c) AVL-insert, into the first AVL tree that is initially empty, the numbers in the array sequentially from the start to the end.

   (d) Initialize the second empty AVL tree.

   (e) Enter a forever while loop to do the following:

      i. Call “findMin” to find the node with the smallest value.

      ii. Call “Remove” to remove the smallest value from the first AVL tree, and display “The process with a priority of %d is now scheduled to run!”

      iii. For the removed value by the i^{th} “Remove”, change it to the value in the i^{th} item of the shuffled array.

      iv. Call “Insert” to insert the removed value (now changed to a different value in the previous step) to the second AVL tree, and display “The process with a priority of %d has run out of its timeslice!”

      v. When the first AVL tree becomes empty, display “Every process has got a chance to run; Please press “Enter” to start the next round!”

      vi. When “Enter” is pressed, swap the two AVL trees, and continue the loop.

Table 1: Performance Measurement

	Round 1	Round 2	Round 3	Round 4	Round 5	Average
AVL						

2 Deliverables

1. Source code
2. Time complexity analysis