CSc 130 Assignment 3
Simulation of A Simple CPU Scheduler Using Binary Heap

Dr. Jinsong Ouyang

1 Requirements

1. Design and implement a Binary Heap class that must support ”insert” and ”deleteMin” 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) Insert, into the first binary heap that is initially empty, the numbers in the array sequentially from the start to the end.
   (d) Initialize the second empty binary heap.
   (e) Enter a forever while loop to do the following:
      i. Collect the first timestamp.
      ii. Call ”deleteMin” to remove the smallest value from the first heap, and display ”The process with a priority of %d is now scheduled to run!”
      iii. For the deleted value by the i^{th} “deleteMin”, change it to the value in the i^{th} item of the shuffled array.
      iv. Call ”Insert” to insert the deleted value (now changed to a different value in the previous step) to the second heap, and display ”The process with a priority of %d has run out of its timeslice!”
      v. When the first heap becomes empty, collect the second timestamp, compute the Δ of the two timestamps, and display ”It took Δ msecs for all processes to run out of their timeslices.”; Please press ”Enter” to start the next round!”
      vi. When ”Enter” is pressed, swap the two heaps, and continue the loop.

Table 1: Performance Measurement

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

2 Deliverables

1. Source code
2. Performance evaluation. Run your program a number of times and record the running time for each round. Then go back to change your 2^{nd} program (if necessary) to collect the same running time information. Represent in a table the performance results by the two solutions, then compare the differences and explain possible causes to the difference in running time.