Lab 10 Report: Min Heap of Integers

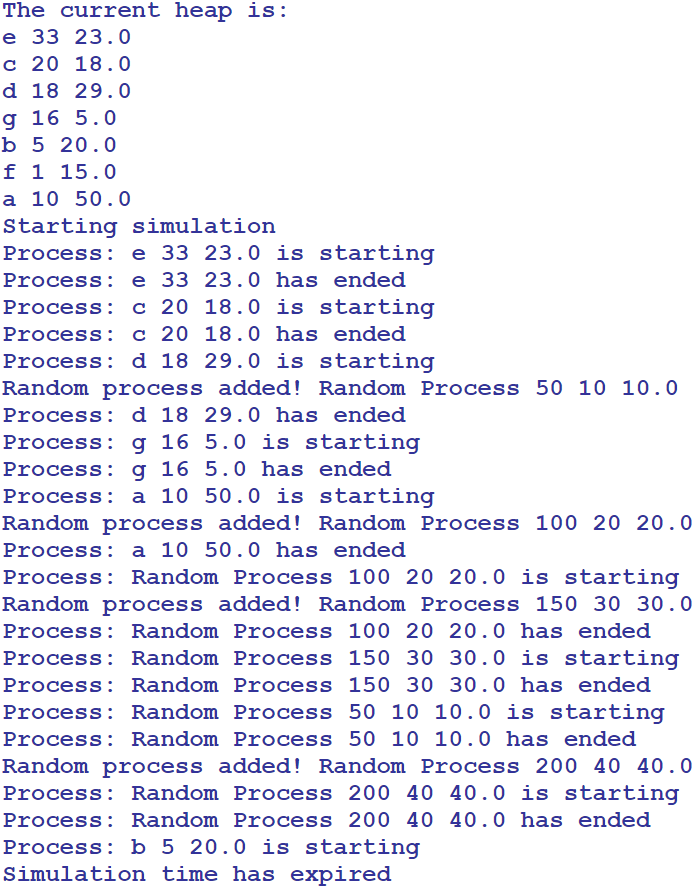
Problem

Create a couple of classes that will simulate a process heap. Download the driver and follow the following instructions!

Proposed Solution

* 1. Create a class called Process
     1. Instance Variables:
        1. Priority: an integer value that corresponds to where it will be in the heap. Needs to be greater than 0.
        2. Time: a double value that corresponds to how long the process will take to run. Needs to be greater than 0.
        3. Name: a string value that is the name of the process
     2. Constructors: Default and Parameterized
     3. Accessors and mutators for each instance variable, make sure to check for valid values.
     4. Methods:
        1. toString: a method that prints out the name, priority, and time.
  2. Create another class called ProcessHeap
     1. Instance Variables:
        1. Heap: an array of type Process which represents the heap
        2. Size: the number of elements in the heap
     2. Constructors: default and parameterized
     3. Other methods:
        1. Insert: This method which returns nothing adds a new element to the heap. It always adds the next element as the next element in breadth order and then bubbles it up based on the priority of the Process.
        2. Peek: This returns the head of the heap
        3. Remove: This returns the root of the heap and then removes it from the heap. IT must then move the last element in the heap (via breadth order) to the root and bubble that value down until it’s in the correct location.
        4. PrintHeap: prints the heap out in breadth order
        5. isEmpty: returns a true or false value for whether or not the queue is empty.
        6. heapSort: Prints out the heap in sorted order. The idea is to remove an element print that element and then repeat that until there are no more elements.

Tests and Results



Problems Encountered

This lab was very straight forward and the only issue encountered was passing in the data from the driver, as we had to figure out what numbers were the process times were and their priority numbers. The console output was different than what was on lab10 for the website but this is because different numbers were inserted.

Conclusions and Discussion

This lab was straight forward and based on examples and code which was shown in class. This gets students prepared and familiar with process heaps, which are very weird as the priority is what places them in the heap. For some reason that was a hard concept for me to grasp during this lab.

Additional Questions

1. What are the major differences between a heap and a binary search tree?

Heaps guarantee that elements on higher levels are greater or small (depending on max or min heap) than elements on lower levels, where the binary search tree does order from left to right for sorted elements.

1. What are some problems that a heap can be used to solve?

Due to the efficiency of the insertion and deletion methods O(log n), they are very useful for querying data for min or max values multiple times. This can be used such as in real world solution, Dijkstra’s algorithm, to figure out the most efficient routing of network packets between two nodes.