October 29, 2019

CSI2110 Assignment II Report

This report explains my solution to assignment 2. It explains the private variables, constructor methods, public methods, private methods, and the implementation of the min and max removal.

Private Variables: The private variables are the Entry buffer, an Entry array for the min heap part and an Entry array for the max heap part, finally two integers to represent the index of the tail element within the min heap and max heap respectively.

Instantiation: The constructor takes an int input that initializers both the min heap and max heap arrays to be equal to the argument. The default is 100 when an argument is not passed into the constructor. The buffer is set to null, representing that there is no Entry in the buffer. The min tail and max tail are set to -1, representing that the max and min heaps are empty.

Public methods:

Size: returns the number of entries in the priority queue currently. It is calculated by added the minTail and maxTail index with 2 to offset zero-indexing with 1 depending if there is an Entry in the buffer.

isEmpty: returns true if the priority queue contains no entries and false otherwise. Determined true if there are no elements in both min and max arrays and if the buffer is empty.

Insert: inserts a given Entry into the priority queue. Insert into buffer if the buffer is empty. Otherwise, it checks to see if the arrays are empty. Then it associates the newly inserted Entry with the Entry in the buffer and calls a helper method insert.

Min: returns the Entry with the minimum key within the priority queue.

Max: returns the Entry with the maximum key within the priority queue.

Print: prints the state of the priority queue.

Private Methods:

Insert: takes two Entries that need to be inserted into the arrays. Inserts the Entry with the greater key into the max heap and the other Entry into the min heap. Then calls the upheap method on both arrays to ensure the maintenance of the heap properties.

upHeap: performs the upheap operation recursively at the specified location. A Boolean argument determines if upheap is to be done for a maxheap or a minheap.

Swap: swaps the entries in the locations passed in as arguments. A Boolean argument determines if the swap if done on the min heap or the max heap.

downHeap: performs the downheap operation recursively at the specified location. A Boolean argument determines if downheap is to be done for a maxheap or a minheap.

Parent: return the location of the parent Entry to a given location.

Implementation of max and min removal:

First the method checks if the priority queue is empty and returns null if it is. Then it checks if the min/max heap if empty (the buffer will have an Entry in this case) and removes and returns the buffer Entry if true. Next, it checks the case that the buffer if empty, but the min/max heap has Entries. In this case, the root of the min/max heap it stored in a temporary Entry variable to be returned and the root is replaced with the tail of the respective heap. Down heap is called starting from the root to maintain the heap properties. The next case it checks is if the key of the buffer Entry is less than/greater than the key of the root of the min/max heap. If true, then the buffer is removed and returned. The final case is if the buffer Entry's key is not less than/greater than the key of the root of the min/max heap. In this case, the operations performed are the same as in the case where the buffer is empty, and the min/max is not empty.