Lab 4:

Implementing Priority Queue Using Heap

- Download the following file
  - Lab4.zip

In lecture you have studied priority queues implementations using sorted and unsorted sequences. We analyzed that both implementations take O(n) for one of the two main operations of the priority queue (insert and removeMin).
By implementing a priority queue using a heap we can execute insert and removeMin in O(log(n)) time reducing time complexity while still storing the Priority Queue in the most efficient amount of space.

The task for this lab involves writing the full implementation of a priority queue using an array implementation for a heap as the data structure. Your TA should review how to implement a heap using an array.
Your TA should also review the upHeap and downHeap operations needed to do insert and removeMin operations, respectively; the main reference for this are the slides on heap priority queues covered in class.

Task details:

1. Review the given PriorityQueue<K,V> interface and Entry<K,V> class.
2. Complete HeapPriorityQueue<K,V> class using a heap, stored as an array of elements Entry<K,V>.
3. Test HeapPriorityQueue<K,V> using HeapPriorityQueueTest.