CMPT 225 D2
Fall 2020
T.Shermer

Assignment 3
Adaptable Heap

Due November 20 at 23:59

You are provided with driving code (Main.cpp) and a skeleton interface (APriorityQueue.cpp) for an adaptable priority queue implemented as a heap. You must implement the methods of the adaptable priority queue:

Entry& insert(k, v)

Takes integer k as the key (priority) and string v as the value and inserts an entry object corresponding to it to the queue. Afterwards, it would return a reference to the entry it just inserted.

Entry& removeMin()

Removes the entry with minimum key and returns a refrence to the new min entry

Entry& min()

Returns a refrence to the minimum entry

void remove(Entry& e)

removes entry e from the queue

void replace(Entry& e, k, v)

replaces the key and value of entry e with k and v and make changes to the priority queue accordingly.

int size()

returns the number of entries in the queue

bool empty()

returns whether if the queue is empty or not *and any private methods that you need.*

min() and removeMin() should throw a QueueEmpty exception (provided in QueueEmpty.h) if called with an emtpy adaptable priority queue. Remove() and replace() should throw a EntryNotFound exception (provided in EntryNotFound.h).

You must also implement the nested class ApriorityQueue::Entry, which is a location-aware entry, as specified in lecture. It has the public methods

int key()

string value()

and any private methods that you need.

Running the compiled program/main file will exercise your class and print some results. You are not allowed to change Main.cpp but you are allowed to change other files. You may also need to create another class for the heap elements; if you do this then it should be a nested class of ApriorityQueue.