

HW 2 Priority Queues

Introduction

In this programming project you are being asked to implement the priority queue data structure in the C++ language. Your implementation should use a Min. Heap data structure. You may use the pseudo-code given in Cormen's text as a guide in your implementation.

The grading of this project will follow the guideline given below:

1. Complete this programming Project (similar to HW2 given in the Fall 2015) semester.
2. Submit the source code as a single zip file to the Instructor by email (alee@syr.edu) by Mar 21 or earlier.
3. The project will be tested via the g++ compiler (version 4.8.4) in Ubuntu (version 14.04).
4. The project will be considered satisfactory if it can be compiled in the above stated environment; can meet the requirement of a majority of test cases and adequate documentation (code are commented and additional explanation are given as a text file named readme.txt when necessary) is provided together in the submission.

The instructor will check the submitted work. If the work is considered as satisfactory, then the instructor will submit the change of grade form (Semester grade will be changed from B- to B) by Mar 31. If the work is not satisfactory, then the semester grade will remains unchanged.

Description

In this project, all data are records with two fields: key (of type int) and value (of type int). You may assume that no two data items has the same key. Priority among the data are defined by via their key values. The smaller the key values, the higher the priority. In precise terms, the Heap data structure under study is a min-heap. The main purpose of this project is to help the student to understand the use of the heap data structure to develop a priority queue (abbreviated as PQ) and the proper functioning of each of the following operations:

1. Insert: insert a data item, say d, to PQ
2. Minimum: Return the data item in PQ with the minimum key (highest priority)
3. Decrease-key: decrease the value of an data item's key to a new value
4. Extract-min: Remove and return the data item with the smallest key value

As as reference, note that an analogue of min-heap (max-heap), was discussed in Cormen, Chapter 6 Section 5.

Design

Develop the priority queue data structure via a min-heap, together with a test program which will be able to test the functionalities of each of the operations (1 to 4). For each of these operations, develop *at least three test cases*.

State the purpose and rationale of each of these test cases and the expected outcome and put them neatly in a text file named `readme.txt`. You may include any other relevant information in the `readme.txt` file to illustrate, explain and support your ideas regarding the implementation submitted.

Implementation

Your implementation should be in the C++ language. In particular, it is expected that your submission will have the following classes developed:

1. MinHeapPQ (Min Heap Priority Queue) with all the operations (1-4, as stated above) implemented
2. Data: represent the data records in this project.

Note:

1. You may add any other members to the above classes if you desire. Your implementation can certainly include other classes you find helpful.
2. You are allowed to use the vector class in C++ Standard Template Library. Do not use any other classes and algorithms given in STL.