Homework 6: Priority Queue and Heap

<PriorityQueue>

**Bài 1**. Tạo giao diện các phần tử và hàng đợi ưu tiên PriotityQueueInterface như sau:

**public** **interface** Entry <K,E> {

K getKey(); //K là khóa của phần tử

E getValue(); //E là giá trị phần tử

}

**public** **interface** PriorityQueueInterface<K , E> {

**public** **int** size();

**public** **boolean** isEmpty();

**public** **void** insert(Entry<K, E> entry); //thêm một entry vào PQ

**public** **void** insert(K k, E e); //thêm phẩn tử có key k và giá trị e vào PQ

**public** Entry<k, E> removeMin(); //loại phần tử có giá trị nhỏ nhất

**public** Entry<k, E> min(); //trả về phần tử có key nhỏ nhất

}

* 1. Xây dựng kiểu dữ liệu UnsortedArrayPriorityQueue sử dụng mảng, cài đặt giao diện PriorityQueueInterface đã xây dựng ở trên với lược đồ gợi ý như sau:

**public** **class** UnsortedArrayPriorityQueue<K **extends** Comparable, E> **implements**

PriorityQueueInterface {

**protected** **class** ArrEntry<K, E> **implements** Entry<K, E>{

**K** key;

**E** element;

**public** ArrEntry (**K** k, **E** e){

}

}

ArrEntry<K, E> [] array;

**int** n = 0;

**int** defaultsize = 1000;

}

* 1. Xây dựng kiểu dữ liệu SortedArrayPriorityQueue sử dụng mảng, cài đặt giao diện PriorityQueueInterface đã xây dựng ở trên với lược đồ gợi ý như trong bài 1.1

**public** **class** SortedArrayPriorityQueue<K **extends** Comparable, E> **implements**

PriorityQueueInterface {

}

* 1. Xây dựng kiểu dữ liệu UnsortedLinkedPriorityQueue sử dụng danh sách liên kết, cài đặt giao diện PriorityQueueInterface đã xây dựng ở trên với lược đồ gợi ý như sau:

**public** **class** UnsortedLinkedPriorityQueue<K **extends** Comparable, E> **implements**

PriorityQueueInterface **{**

**protected** **class** NodeEntry<K, E> **implements** Entry<K, E>{

**private** K key;

**private** E element;

**private** NodeEntry<K, E> next;

**public** ArrEntry (K k, E e){

}

}

**private** NodeEntry<K,E> head;

**private** NodeEntry<K,E> tail;

**int** n = 0;

}

* 1. Xây dựng kiểu dữ liệu SortedLinkPriorityQueue sử dụng danh sách liên kết, cài đặt giao diện PriorityQueueInterface đã xây dựng ở trên với lược đồ gợi ý như trong bài 1.3:

**public** **class** SortedLinkedPriorityQueue<K **extends** Comparable, E> **implements**

PriorityQueueInterface{

**}**

* 1. Viết hàm test các kiểu dữ liệu PriorityQueue đã triển khai với:

- Danh sách các số nguyên, giá trị phần tử dùng làm khóa.

- Danh sách các đối tượng có khóa và giá trị khác nhau. Ví dụ: đối tượng *hàng hóa* bao gồm *tên hàng hóa* (giá trị), *giá tiền* (khóa).

<Heap>

**Bài 2**. Xây dựng cấu trúc dữ liệu hàng đợi ưu tiên sử dụng heap cài đặt bằng mảng với lược đồ như sau:

**public** **class** MinHeapPriorityQueue<K **extends** Comparable, E> **extends**

SortedArrayPriorityQueue {

ArrEntry<K,E> heapPQ[];

//Các phương thức bổ sung

**protected** **void** upHeap() //vun lên

**protected** **void** downHeap() //vun xuống

}

**Bài 3.** Sử dụng cấu trúc dữ liệu HeapPriorityQueue để viết hàm HeapSort sắp xếp dãy số theo thứ tự tăng dần.

Chạy và so sánh thời gian thực hiện của 4 thuật toán sắp xếp SelectionSort, HeapSort, QuickSort và MergeSort.