

문제: Priority Queue Simulator

문제 설명 :

Heap (max heap) 구조를 이용하여 priority queue의 동작을 흉내 내는 간단한 프로그램을 작성하고자 한다. 입력으로 key 값이 들어오면 수업시간에 배운 대로 insertion 알고리즘을 이용하여 heap에 삽입한다. 삭제(delete) 명령이 오면 heap의 root 노드를 삭제하고, 제일 마지막에 있는 key 값을 root에 올린 후, adjust 알고리즘으로 heap을 재구성한다. 삽입과 삭제가 반복되는 동안 heap 구조가 제대로 유지되고 있는지를 점검하기 위해 중간 중간에 heap의 내용 일부를 출력하는 명령도 있다. Heap 구조를 점검하기 위한 명령이 들어오면 구성된 heap에서 root, left son(root), left son(left son(root)), left son(left son(left son(root)))...를 출력한다. 즉, heap 구조에서 root에서 출발하여 좌측 자식을 따라가며 key 값들을 출력한다.

입력으로는 각 줄에 key 값 또는 아래의 명령어 중 하나가 들어오며, 그 명령의 의미는 설명과 같다.

key: Heap에 key 값을 삽입한다. 삽입 시 insertion 알고리즘을 이용한다. 즉, heap 마지막에 key 값을 넣은 후, 위로 올라가면서 key 값의 위치를 조정하여 heap을 재구성한다.

d (delete): Heap에서 root를 삭제한다. 삭제 후, 마지막 key값을 root에 올린 후, adjust 알고리즘을 이용하여 heap을 조정한다.

r (report): Heap 구조의 일부를 보여준다. 앞에서 설명했듯이 heap 구조의 root에서 출발하여 좌측 자식을 따라가며 key 값들을 보여준다.

q (quit): 모든 작업을 종료한다.

【입 력】

입력파일의 이름은 queue.inp 이다. 입력 파일의 각 줄에는 key 값 또는 한 글자로 된 명령어가 들어온다. Key 값은 양의 정수이며, 삽입되는 key의 개수는 최대 1,000,000개 이다. 동일한 값을 가지는 key는 입력되지 않는다.

【출 력】

출력 파일의 이름은 queue.out이다. 주어진 입력에서 'r' 명령이 나올 때 마다 그 때의 heap 구조를 보여준다. 이때 앞에서 설명했듯이, root부터 좌측 자식을 따라가며 key 값들을 한 줄에 보인다. 각 값들은 공백 문자로 구분된다.

【실행 예】

입력 예	입력 예에 대한 출력
111	123 111 12
123	563 42 12
99	765 563 42 12
12	
9	
r	
321	
332	
42	
d	
d	
d	
563	
324	
r	
765	
r	
q	

제한조건: 프로그램은 queue.{c,cpp,java}로 한다.