

## 제6회 연습문제

1. heapsort 알고리즘을 구현하여 지난 과제에서처럼 다른 알고리즘들과 성능을 비교하라.
2. 우선순위 큐를 구현하는 naive한 방법의 하나는 그냥 정렬되지 않은 배열을 사용하는 것이다. 새로운 원소는 항상 배열의 맨 끝에 삽입하고, 최대값 삭제는 최대값을  $O(N)$  시간에 찾아서 삭제하고 배열의 마지막 원소를 최대값이 있던 위치로 이동하여 배열의 중간에 빈 칸이 없도록 처리한다. heap을 이용하여 우선순위큐를 구현한 후 이렇게 단순하게 배열로 구현된 경우와 실행속도를 비교하라. 성능의 비교는 다음과 같이 한다. 우선  $N$  개의 0에서  $N$ 사이의 정수를 랜덤하게 생성하여 우선순위큐에 삽입한다. 그런 다음  $M$ 번의 삽입 혹은 최대값 삭제 연산을 연속하여 실행하는데 걸리는 총 시간을 측정한다.  $M$ 번의 연산 각각은 우선  $1/2$ 의 확률로 삽입 연산인지 혹은 최대값 삭제 연산인지를 결정한 후, 삽입 연산인 경우 다시 0에서  $N$ 사이의 정수를 랜덤하게 생성하여 삽입할 데이터를 결정한다.

```
static void test(int N, int M) {  
    for (int i=0; i<N; i++)  
        pqueue.add(rd.nextInt(N)); // pqueue is the priority queue to test  
  
    for (int i=0; i<M; i++) {  
        if (rd.nextInt(2)==0) // add  
            pqueue.add(rd.nextInt(N));  
        else if (!pqueue.empty()) // extract max  
            pqueue.extractMax();  
    }  
}
```