프로그래밍 과제 05

1. 지난 주의 과제 1번에 마지막 2가지 정렬 알고리즘을 추가하여 완성하라.

2. 주소록 파일이 있다. 파일의 한 라인은 한 사람에 대한 이름, 회사(company), 주소(address), 우편번호(zip code), 전화번호, 이메일 주소의 6가지 필드로 구성된다. 필드와 필드는 하나의 "|" 문자로 구분되어 있다. 샘플파일을 다운로드하여 사용하라. 이 주소록 파일을 읽은 후 각각의 사람을 하나의 객체로 저장한 후 (C 프로그램의 경우 각각의 사람을 하나의 `struct`로 표현한다) 다음과 같이 사용자의 명령을 수행하는 프로그램을 작성하라. 단, 반드시 Java API, 혹은 C, C++ 표준 라이브러리가 제공하는 정렬 기능을 사용하라. 즉 정렬 알고리즘을 직접 구현해서는 안된다. 불필요한 공백이나 탭(tap) 문자 등이 저장되지 않도록 주의하라.

```
$ read address.txt
$ sort by name    // 사람들을 name의 알파벳 순으로 정렬한다.
$ print           // 정렬된 순서대로 모든 사람에 대한 모든 정보를 다음과 같이 화면에 출력한다.
Abel
    Company: Rangoni Of Florence
    Address: 37275 St  Rt 17m M Middle Island Suffolk NY
    Zipcode: 11953
    Phones: 631-335-3414
    Email: amaclead@gmail.com
Adelina
    Company: Courtyard By Marriott
    Address: 80 Pittsford Victor Rd #9 Cleveland Cuyahoga OH
    Zipcode: 44103
    Phones: 216-230-4892
    Email: adelina_nabours@gmail.com
…
$ sort by address       // 사람들을 주소의 알파벳 순으로 정렬한다.
$ print           // 정렬된 순서대로 모든 사람에 대한 모든 정보를 위와 동일한 형식으로 출력한다.
…
$ exit
```

3. 우선순위 큐를 구현하는 단순한 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. 이렇게 단순하게 배열로 구현한 경우와 이진 힙을 이용하여 우선순위 큐를 구현 한 경우의 실행속도를 비교하라. 비교는 다음과 같이 한다. 우선 N 개의 0에서 N사이의 정수를 랜덤하게 생성하여 우선순위큐에 삽입한다. 그런 다음 M번의 삽입 혹은 최대값 삭제 연산을 연속하여 실행하는데 걸리는 총 시간을 측정한다. M번의 연산 각각은 우선 1/2의 확률로 삽입 연 산인지 혹은 최대값 삭제 연산인지를 결정한 후, 삽입 연산인 경우 다시 0에서 N사이의 정수를 랜덤하게 생성 하여 삽입할 데이터를 결정한다. N과 M의 값은 프로그램 시작시 키보드로 부터 입력받는다. 두 방법에 대해서 완벽하게 동일한 데이터로 테스트할 필요는 없고, 각각에 대해 따로 랜덤 데이터를 생성하여 테스트하면 된다.

```
void test(int N, int M) {
    for (int i=0; i<N; i++)
        pqueue.add(rd.nextInt(N));      // pqueue는 테스트할 우선순위 큐

    for (int i=0; i<M; i++) {
        if (rd.nextInt(2) == 0 || pqueue.empty() )   // add
            pqueue.add(rd.nextInt(N));
        else                                         // extract max
            pqueue.extractMax();
    }
}
```