

프로그래밍 과제 01

1. 프로그램을 시작하면 먼저 데이터 파일 `data.txt`를 읽는다. 파일의 맨 앞에는 입력될 정수의 개수 $N \leq 10000$ 이 주어지고, 이어서 N 개의 정수들 a_0, a_1, \dots, a_{N-1} 이 주어진다. 그런 다음 사용자에게 다음과 같은 기능을 제공한다. 사용자가 음이 아닌 두 정수 i 와 j 를(단, $i \leq j$) 입력하면, a_i, a_{i+1}, \dots, a_j 중에서 최소값을 찾아서 출력한다. 이렇게 사용자가 원하는 구간에서 최소값을 찾는 연산을 range query라고 부르자. 사용자가 -1 을 입력할 때 까지 반복해서 range query를 수행하는 프로그램을 작성한다. 예를 들어서 다음과 같이 작동해야 한다. $N = 14$ 이고, 정수들은

12, 9, 1, 4, 10, 1, 7, 8, 9, 3, 6, 11, 8, 2

이라고 가정하자.

```
2 5    // 사용자가 i=2, j=5를 입력하였다. 즉, 1, 4, 10, 1 중에서 최소값을 찾으라는 의미이다.
1      // 최소값 1을 출력한다.
6 9    // i=6, j=9, 즉, 7, 8, 9, 3중에서 최소값을 찾아야 한다.
3      // 최소값 3을 출력한다.
5 5    // i=5, j=5이다.
1      // 최소값 1을 출력한다.
-1     // 프로그램을 종료한다.
```

매 번 단순히 주어진 구간에서 최소값을 찾을 경우 한 번의 query를 처리하는데 필요한 최악의 시간복잡도는 $O(N)$ 이 될 것이다. **왜 그런가?**

이런 종류의 반복적인 query를 수행하는 프로그램에서는 query를 받기 전에 미리 적절한 사전 처리(pre-processing)를 해 둬으로써 각 query의 처리시간을 줄일 수 있다. 다음은 가장 기본적인 알고리즘의 하나이다.

$M = \lfloor \sqrt{N} \rfloor$ 이라고 하자. 주어진 N 개의 정수들을 연속된 M 개씩 그룹핑하여 각 그룹에서의 최소값을 미리 찾아 저장해 둔다. 위의 예에서는 $M = 3$ 이므로 14개의 정수들은 다음과 같이 5개의 그룹으로 분할되고, 각 그룹에서의 최소값 m_0, m_1, \dots, m_4 는 아래와 같을 것이다.

그룹 분할: [12,9,1], [4,10,1], [7,8,9], [3,6,11], [8,2]
 각 그룹의 최소값: $m_0 = 1, m_1 = 1, m_2 = 7, m_3 = 3, m_4 = 2$

이렇게 미리 찾아 놓은 각 그룹의 최소값을 이용하여 다음과 같이 사용자가 원하는 구간의 최소값을 계산한다. 예를 들어 $i = 1, j = 10$ 이라고 하자. a_1 은 첫 번째 그룹에 속하고, a_{10} 은 4번째 그룹에 속한다. 두 번째 그룹과 세 번째 그룹의 최소값은 각각 $m_1 = 1, m_2 = 7$ 로 이미 구해져 있으므로, 우리는 $a_1, a_2, m_1, m_2, a_9, a_{10}$ 중에서 최소값을 찾으면 된다. 이 값들의 개수는 최악의 경우 몇 개인가? 이 알고리즘을 구현하라. 이 알고리즘에서 하나의 query 처리에 필요한 최악의 경우 시간복잡도는 얼마인가?

입력 파일의 예: `data.txt`

입출력 예: (최소값의 배열 index는 출력하지 않아도 된다.)

i and j	최소값	i and j	최소값
0 9999	data[2463] = 5	6746 9872	data[9164] = 34
1233 8723	data[2463] = 5	8765 9000	data[8862] = 43
3674 4782	data[4155] = 60	6543 7654	data[6816] = 55
5368	data[211] = 89	1234 2134	data[2134] = 64