

Sorting

[문제] 주어진 수열을 정렬하는 방법은 여러가지가 있다. 다음 3가지 방법을 이용하여 수열을 오름순으로 정렬한다.

- 1) 삽입정렬(insertion sort) : 주어진 수열에서 맨 앞에 있는 원소가 정렬되어 있다고 하고 2번째 원소부터 정렬된 수열에서 오름차순으로 자기 위치를 찾아간다.
- 2) 선택정렬(selection sort) : 주어진 수열에서 가장 작은 값을 찾아 첫번째 원소와 교체한다. 이것을 패스(pass) 혹은 단계라 한다. 다음 패스는 정렬되지 않은 나머지 수열에 대해서 같은 방식으로 적용한다. 정렬되지 않은 수열이 있으면 정렬이 완료될 때 까지 재귀적으로 적용한다.
- 3) 힙정렬(heap sort) : 수열을 먼저 힙(배열)에 배치한 다음, 이를 힙의 성질(min tree)을 활용하여 최소값이 root에 오도록 함)을 만족하도록 heap 구축과정을 실행한다. 이 과정이 완료되면 힙의 초기 상태가 된다. 이 힙에서 root의 값을 출력하면 수열의 최소값을 얻을 수 있다. root에 원소가 삭제되면 다시 힙의 성질을 만족하지 못하므로 힙을 재구축해야 한다. 이를 updateHeap()이라 한다. 이 과정은 수열이 정렬이 완료될 때까지 재귀적으로 적용된다.
- 4) 퀵정렬(quick sort) : 주어진 수열의 맨 앞에 오는 원소를 피벗(pivot)으로 한다. 이 pivot를 기준으로 작은 값은 왼쪽배열로 pivot보다 큰 값은 배열의 오른쪽에 온다. 이런 수행을 위해서 수열의 앞쪽에서 뒷쪽으로 이동하면서 pivot보다 큰 값을 선택하고, 뒷쪽에서 앞쪽으로 오면서 pivot보다 작은 값을 찾아서 앞에서 찾은 값과 교환한다. 더 이상 교환할 원소가 없으면 pivot과 pivot 보다 작은 값 중에 가장 오른쪽에 있는 원소와 바꾸어 pivot의 제자리를 찾는다. 여기까지가 하나의 패스가 된다. 분할된 리스트에 대해서는 이전의 pivot보다 작은 값을 가지는 리스트 먼저 수행한다.

이 과제에서는 정렬되고 있는 중간단계가 제대로 동작하고 있는지 점검하고자 한다.

(1)삽입정렬에서 k 번째(여기서 수열에서 두번째 원소가 $k = 1$ 이다) 원소가 자기 위치를 찾을 때까지의 총 비교횟수를 출력한다. (2) 선택정렬에서는 k 번째 pass(단계)가 끝났을 때의 주어진 수열의 순서를 출력한다. (3) 힙정렬에서는 k 개의 숫자가 정렬된 후의 힙에 저장되어 있는 순서를 출력한다. (4) 퀵정렬은 k 번째 pivot이 제자리를 찾았을 때의 수열을 출력한다.

[입출력] 입력의 첫 줄에는 정렬방법과 pass 단계가 주어진다. 정렬방법은 1 - 삽입정렬, 2 - 선택정렬, 3 - 힙정렬, 4 - 퀵정렬이다. k ($k > 0$) 값은 앞 문제에서 설명한 각 정렬방법별 원소의 수, 또는 단계 혹은 k 번째 정렬후의 중간단계를 나타낸다. 두번째 줄에는 정수의 개수 N ($5 \leq N \leq 1000$)이 주어진다. 그리고 이어지는 줄에는 N 개의 정수가 온다.

[예제]

입력 stdin	출력 stdout
<pre>1 1 //삽입정렬 (1), k=1 5 //N=5 총 정수수 16 2 7 -10 1</pre>	<pre>1 //2가 자기 위치를 찾을 때까지의 총 //비교횟수</pre>
<pre>2 1 //선택정렬 (2), k=1 5 //N=5 총 정수수 16 2 7 -10 1</pre>	<pre>-10 //선택정렬 첫번째 단계 완료, 즉 2 //가장 작은 수를 찾아서 첫번째 원 7 //소와 교환한 중간 결과 16 1</pre>
<pre>3 1 //힙정렬 (3), k=1 5 //N=5 16 2 7 -10 1</pre>	<pre>1 //가장 작은 수가 정렬되고 난 뒤의 2 // 힙에 남아 있는 수열의 순서 7 16</pre>
<pre>4 1 //퀵정렬 (4), k=1 5 //N=5 16 2 7 -10 1</pre>	<pre>1 // pivot 16이 제자리 온 뒤의 2 // 수열의 순서 7 -10 16</pre>

[제한조건] 프로그램의 이름은 pa01_sort.{py,c,cpp,java}이다. 제출 횟수는 최대 15번이며 허용 시간은 데이터 당 제한 시간은 1초, 허용가능 코드의 최대 크기는 10,000 bytes 이다. 문제 풀이 마감시간은 2023년 3월 16일 **24:00** 이다. 제출한 프로그램에 대한 풀이(방법과 코드설명)를 작성하여 2023년 3월 17일 24:00까지 업로드(NESPA 풀이 게시판) 해야 한다. 업로드한 프로그램 풀이과정은 마감이 지나면 공개된다.