

## 숙제 1. 정렬과 $k$ -nearest neighbors

### ○ 문제 설명

현재 위치에서 가장 가까운  $k$ 개의 주유소들의 좌표값과 거리를 출력

### ○ 입력: 파일 이름 (파일에는 다음 내용이 차례대로 저장되어 있음)

- 현재 위치의  $x, y$  좌표 ( $0.0 \leq x, y \leq 1000.0$ )
- $k$  값 (정수. -1이면 전체 선택)
- 주유소 좌표의 수  $n$  (정수)
- $n$ 개의  $(x, y)$  좌표쌍이 실수로 저장

### ○ 구현 방법

1. (기본적인 방법) 모든 주유소를 거리의 오름차순으로 무조건 정렬한 후, 거리가 가장 가까운  $k$ 개를 출력
2. (개선된 방법)  $k$ 의 크기에 따라 다른 알고리즘들을 사용

병합

$k=N \rightarrow \text{TIMsort}$ ,

$k<N \rightarrow \text{MAX Heap}$

### ○ 점수: 기본적인 방법(10점) + 개선된 방법(20점)

- 성능은 알고리즘의 실행 시간(System.currentTimeMillis())으로 평가
- 모든 정렬 알고리즘은 직접 구현 (Java의 정렬 관련 API들은 사용 금지)
- 실행 시간을 계산할 때, 입출력 시간은 포함시키지 말 것!

### ○ 과제 제출 방법: **HW1.java** 하나의 파일만 제출

- ① public class HW1 (파일 내에 나머지 클래스들은 public이 아님)
  - ② default package 사용
  - ③ 첫 번째 줄에는 자신의 학번과 이름을 주석으로 추가. 그 외 주석은 모두 삭제
- ← 위의 조건들을 하나라도 만족하지 않는 과제는 심사하지 않음!!

### ○ 실행의 예

```
파일 이름? 1m.txt
k = 3일 때의 실행시간 = 25ms
0: (500.643161, 500.518567)  거리 = 0.826177
1: (499.320806, 499.487212)  거리 = 0.851033
2: (499.282574, 500.461823)  거리 = 0.853218
```

```

파일 이름? 1m.txt
k = 10일 때의 실행시간 = 45ms
0: (500.643161, 500.518567) 거리 = 0.826177
1: (499.320806, 499.487212) 거리 = 0.851033
2: (499.282574, 500.461823) 거리 = 0.853218
3: (500.447045, 499.036355) 거리 = 1.062291
4: (498.932106, 499.894260) 거리 = 1.073116
5: (499.240642, 500.867269) 거리 = 1.152728
6: (498.826456, 501.174035) 거리 = 1.659989
7: (500.705158, 501.549838) 거리 = 1.702717
8: (499.726337, 501.684231) 거리 = 1.706319
9: (501.618603, 499.197198) 거리 = 1.806755

```

```

파일 이름? 1m.txt Timsort
k = 1000000일 때의 실행시간 = 531ms
0: (500.643161, 500.518567) 거리 = 0.826177
1: (499.320806, 499.487212) 거리 = 0.851033
2: (499.282574, 500.461823) 거리 = 0.853218
3: (500.447045, 499.036355) 거리 = 1.062291
...

```

#### ○ Java에서 파일 입력 방법

```

Scanner sc = new Scanner(System.in);
System.out.println("입력 파일 이름? ");
String fname = sc.nextLine();
sc.close();
try {
    sc = new Scanner(new File(fname));
    int input_data = sc.nextInt(); // file에서 정수를 읽어 input_data에 저장
                                // 이후 sc.nextInt()를 이용하여 계속 정수 입력
} catch (IOException e) { System.out.println(e); return; }

if (sc != null) sc.close();

```