

21000549-003 알고리즘
Programming Assignment #2
Due: 2020년 10월 16일 23:59PM

좌표평면에 있는 유한 개의 점들 중 최단거리인 쌍 구하기

1. Mission Objective

유한 개의 점 $P_0, P_1, P_2, \dots, P_{n-1}$ 이 있다. 이들 중 가장 거리가 가까운 두 점 P_i 와 P_j 를 찾아서 (단, $0 \leq i < j < n$), 그 두 점의 거리를 반환하는 함수를 만든다.

2. 파일 및 함수 설명

여러분의 리눅스 계정에 **pa1** 디렉토리가 있고, 그 안에 다음과 같은 파일들이 있을 것이다.

pa2.java : **main** 함수가 있는 소스코드 파일로, **closest.java** 에 있는 소스코드를 실행할 때 사용하면 된다. 이 파일의 내용은 점수에 아무런 영향을 미치지 않으므로 마음껏 편집해도 된다 (심지어 컴파일이 되지 않아도 됨)

closest.java : 최단거리의 두 점의 거리를 구하기 위한 클래스가 선언된 파일. 각 멤버 변수 및 멤버 함수의 의미는 다음과 같다.

- **class point** : 평면상의 점 정보를 저장하기 위한 클래스
- **pnt[]** : 점들의 배열
- **closest()** : 생성자.
- **setPoints(point p)** : **pnt**를 정해주는 함수로, **p**에는 멤버 변수 **pnt**에 들어갈 배열이 들어간다.
- **getMinDist()** : 점들 중 최단거리인 쌍의 거리를 구하는 함수로, 여러분이 작업해야 할 곳이다.

이외에도 멤버 변수나 멤버 함수가 더 필요하면 선언 및 정의해서 써도 된다.

(**point** 클래스, 생성자, **setPoints** 함수는 절대로 변경하면 안 됨)

score.o : 채점을 위한 파일. 수정이나 삭제 금지.

3. 실시 예

컴파일은 다음과 같이 한다:

```
$ javac closest.java
$ javac pa2.java
```

실행은 다음과 같이 한다:

```
$ java pa2
```

수정하지 않았을 경우, inversion의 개수를 구하는 프로그램을 잘 만들었다면, 실행했을 때 **Minimum distance: 1.4142135623730951**이 떠야 할 것이다.

4. 채점 방식

프로그래밍 후, 여러분이 점수를 예측하기 위해서는 `score.class` 파일을 활용하면 되며,

```
$ javac closest.java
$ java score
```

채점은 1) 알고리즘의 정확성 평가 2) 특수한 경우에도 가능한지 평가 3) Stress test로 구성되어 있으며, 각각 2, 1.2, 4.8점이다. 모든 테스트가 성공적으로 끝나면 8점을 보장받게 된다. 그 중 stress test에서는 점의 개수 100만개가 될 때까지 테스트를 해 보며, 모든 테스트가 각각 0.3초 안에 끝나야 8점을 받게 된다.

채점 후, 8점 받은 학생들끼리 수행 시간을 비교해서 빠른 순서대로 최종적으로 10점부터 7점까지 부여할 예정임.

5. 제출

- 10월 16일 금요일 23:59까지. 시계는 제출용 서버의 시계를 따른다.

- 여러분의 계정에 **PA2** 디렉토리가 있고, 이 안에 다음과 같은 파일들이 있을 것이다:

```
closest.java
pa2.java
score.class
```

이 중, `closest.java` 파일의 최종 수정 시각으로 delay 여부를 판별할 것임.

- 만일 위의 파일들 중 어느 하나를 지웠으면 담당 교수에게 문의할 것.

- 하루 Delay 되면 본인 획득 점수의 반을 받게 되며, 이를 이상 Delay 되면 0점.

- Copy는 해당 PA 0점. 문제 해결 아이디어를 내기 위해서는 어떤 것을 참조해도 상관 없지만, 소스코드를 찔 때는 다른 사람 것이나 웹사이트를 보면 안 됨. 반드시 본인이 프로그램을 설계하고 타이핑 하여 만들어야 함.