

21000549-001/002 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.cpp: **main** 함수가 있는 소스코드 파일로, **closest.cpp**에 있는 소스코드를 실행할 때 사용하면 된다. 이 파일의 내용은 점수에 아무런 영향을 미치지 않으므로 마음껏 편집해도 된다 (심지어 컴파일이 되지 않아도 됨)

point.h, point.cpp: 평면상의 점 정보를 저장하기 위한 클래스

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

- ***pnt**: 점들의 배열
- **num**: 점들의 개수
- **closest(void)**: 생성자.
- **~closest(void)**: 소멸자
- **setPoints(point* p, int n)**: **pnt**와 **num**의 값을 정해주는 함수로, **p**와 **n**에는 각각 멤버 변수 **pnt**에 들어갈 배열과 그 배열의 길이가 들어간다.
- **getMinDist(void)**: 점들 중 최단거리인 쌍의 거리를 구하는 함수로, 여러분이 작업해야 할 곳이다.

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

closest.cpp: **closest.h**에서 선언된 메소드들을 정의해놓은 파일. 이 중에서 **double closest::getMinDist(void)**의 내용을 편집하면 된다. 또한, **closest.h**에서 메소드를 추가로 선언했다면, **closest.cpp**에서 추가로 정의해줘야 한다. (생성자, 소멸자, **setPoints** 함수는 절대로 변경하면 안 됨)

score.o: 채점을 위한 파일. 수정이나 삭제 금지.

Makefile: **make** 명령을 위한 매크로 파일. **make**라는 명령을 내리며 **run**이라는 실행파일이 만들어진다. 이들을 지우고 싶으면 **make clean** 명령을 내려주면 된다.

3. 실시 예

make 명령 후, **main.cpp**를 수정하지 않았을 경우, **inversion**의 개수를 구하는 프로그램을 잘 만들었다면, 실행했을 때 **Minimum distance: 1.41421**이 떠야 할 것이다.

4. 채점 방식

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

```
$ make score
```

이와 같이 컴파일하면 `score`라고 하는 실행파일이 만들어진다. 이는 `main.cpp`와는 무관하게 만들어지며, 실행하게 되면 여러분들이 받게 될 점수가 출력된다. `make clean` 명령을 내리면 이 파일 역시 지워진다.

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 1초 안에 끝나야 8점을 받게 된다.

채점 후, 001, 002반 전체 학생들 중에서 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.cpp`
`closest.h`
`pa2.cpp`
`point.cpp`
`point.h`
`score.o`
`Makefile`
이 중, `closest.h` 파일과 `closest.cpp` 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.