

### 1. 알고리즘 구성

정수를 입력 받아 점의 개수를 결정하고 범위를 입력받아 랜덤으로 점을 개수에 맞게 범위 내에 생성한다. 그리고 dist2 함수를 사용하여 거리의 제곱을 계산하고 이를 비교하여 최소값과 최댓값을 구한다.

### 2. 코드 분석 및 작동원리

입력이 들어오게 되면 가장 큰 값과 작은 작은 값을 저장해 둔 후 계속하여 새로운 값들과 비교하여 기록을 갱신한다. 단 점이 2개 미만이면 거리 비교가 불가능 하므로 오류 처리를 해주었다. Points 객체를 생성하여 x 좌표와 y 좌표의 증분을 구한 후 이를 제곱한 값을 더해 거리의 제곱 값을 구하였다. 단순 비교만 하면 되기에 루트는 취하지 않았다. 그리고 이 값들을 비교하여 최소와 최댓값을 저장하여 갱신하는 방식으로 구하였다.

### 3. 실행 결과

```
jy@jy:~/intern_ws/day1_hw2/build$ make
Consolidate compiler generated dependencies of target test
[ 33%] Building CXX object CMakeFiles/test.dir/src/Points.cpp.o
[ 66%] Linking CXX executable test
[100%] Built target test
jy@jy:~/intern_ws/day1_hw2/build$ ./test
점 개수 입력: 8
x 범위 [xmin xmax]: 1 9
y 범위 [ymin ymax]: 3 8
P0 = (9,4)
P1 = (5,4)
P2 = (9,7)
P3 = (6,7)
P4 = (3,3)
P5 = (3,6)
P6 = (8,6)
P7 = (9,3)
최소 거리 점쌍: P0(9,4) - P7(9,3)
제곱거리 = 1
최대 거리 점쌍: P2(9,7) - P4(3,3)
제곱거리 = 52
```

사진과 같이 입력한 개수에 따라 점이 랜덤으로 범위 내에 생성되고 최소 거리와 최대 거리(제곱) 알려준다.