

## 고급 소프트웨어 실습

분반: 4 분반

학번: 20192133

이름: 김아영

과제 2. 실습 문제 1 에서 간단한 Delaunay triangulation 에 대하여 살펴보았다. 임의의 개수의 포인트 셋에 대한 알고리즘을 수도 코드 형태로 작성해 보시오.

Points // 임의의 개수의 포인트 셋

// 세 점의 외접원의 중심과 반지름을 구하는 함수

Function getCenter(point1, point2, point3):

1. 세 점을 이용하여 외심을 구하기 위해 다음 연립방정식의 해 구하기

Point1 = (x1, y1), point2 = (x2, y2), point3 = (x3, y3)

$$\begin{bmatrix} C_x \\ C_y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}^{-1} \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ x_3^2 - x_2^2 + y_3^2 - y_2^2 \end{bmatrix}$$

2. 외심과 점 사이의 거리를 통해 외접원의 반지름 구하기
3. Return 외심, 반지름

//point 가 중심이 center 이고 반지름이 radius 인 점 내부에 있는지 확인하는 함수

Function isIncluded(point, center, radius):

1. Point 와 center 사이에 거리 distance 계산
2. Distance 와 radius 비교
  - A. If distance 가 radius 보다 작다면 return True
  - B. Else if distance 가 radius 보다 크다면 return False

// 임의의 개수의 포인트셋에 대한 가능한 3 개의 점들의 조합들을 구하는 함수

Function getCombinations(points):

Combinations // 조합들

1. DFS 를 통해 가능한 3 개의 점들의 조합을 구하여 combinations 에 저장
2. Return Combinations

Combinations = getCombinations(points)

Result // 들로네 삼각분할 결과

For combination in combinations:

center, radius = getCenter(combination[0], combination[1], combination[2])

For point in points:

1. isIncluded(point, center, radius) 함수를 통해 point 가 해당 외접원에 있는지  
확인
  - A. 내부에 있다면 break
1. 해당 외접원 내부에 다른 점이 있지 않다면 Result 에 해당 combination 추가