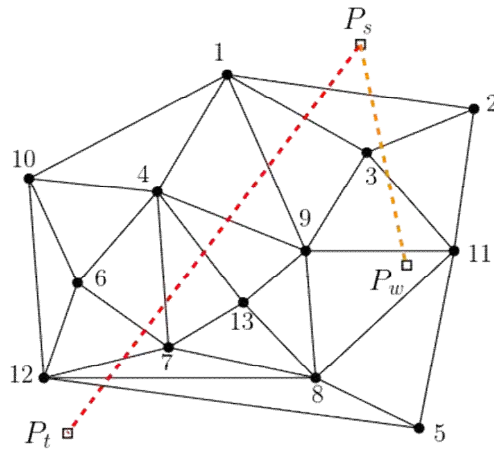


Delaunay Path

[문제] 우리는 주어진 2차원 점 집합 S 에 대하여 Delaunay Triangulation을 구한다. 이 결과 전체 구역은 Delaunay triangle로 Decompose된다. 우리는 이 구역의 두 점을 연결하는 직선 path가 교차하는 Delaunay Triangle의 순서를 찾으려고 한다. 즉 어떤 robot이 직선으로 진행할 때 만나는 DT의 순서를 찾으려는 것이다. 지나가는 face는 이 face를 이루는 삼각형의 정점 vertex를 리스트 [a, b, c] 로 출력한다.

[입력] points.txt 파일의 첫 줄에 점의 수 N 이 주어진다. 그리고 이어지는 N 개의 줄에 각각 점 p_i 의 좌표가 integer x_i, y_i 으로 주어진다. 단 $1 \leq x_i, y_i \leq 1000$ 이다. 그리고 마지막 줄에 두 점 $P_s = (x_s, y_s), P_t = (x_t, y_t)$ 의 좌표가 4개의 정수 x_s, y_s, x_t, y_t 로 주어진다.



만일 위와 같으면 지나가는 face의 순서는 [1, 3, 2], [1, 9, 3], [1, 4, 9], [4, 13, 9], [4, 7, 13], [4, 6, 7], [6, 12, 7], [12, 8, 7], [12, 5, 8]가 되어야 한다. 단 점은 $CH(\{p_i\})$ 의 내부외부 모두 가능하다. 만일 P_w 가 끝점이라면 그 답은 [1, 3, 2], [2, 3, 11], [3, 9, 11], [9, 11, 8]이 되어야 한다. 만일 직선이 face가 아니라 하나의 vertex를 정확하게 지나면¹⁾ 점 번호 하나만을 출력해야 한다. face의 index를 출력하는 순서는 Delaunay triangle을 구한 뒤 얻어지는 simplices의 순서를 그대로 사용한다(반시계방향).

[조건] points.txt의 점의 개수는 최대 100개 이하이다. 4월 30일 저녁 10시까지 ESPA로 십습 보고서 게시판으로 제출하면 된다. 제출 내용은 NAME_DTpath.py이다. 단 그림을 그려서 답이 올바른지를 확인할 수 있도록 해야 한다. 예를 들어 만나는 삼각형의 edge를 다른 색, 굵기로 표시한다든지 하면 더 좋다. 자신이 사용한 데이터를 2종 이상 공개한다. 이것으로 다른 학생의 코드를 검사한다. NAME_dt{01,02}.txt로 한다.

1) 일종의 degenerated case이다.