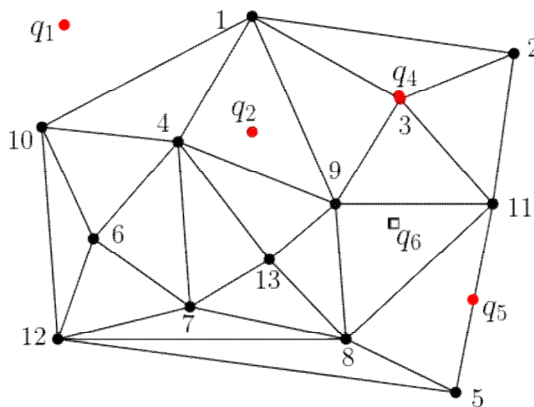


# Delaunay Location

[문제] 2차원 점 집합  $S=\{s_i\}$ 에 대하여 들로니 삼각분할 (Delaunay Triangulation),  $DT(S)$ 가 있다. 이  $DT(S)$ 에 대하여 질문 점  $q_i=(x_i, y_i)$ 에 대하여 이 점을 포함하고 있는 delaunay triangle, 또는 outer region을 판정한다. 이  $q_i=(x_i, y_i)$ 를 포함하고 있는 기하객체는 triangle face, edge, vertex, outer face 중 하나이다. 각각의 경우에 각각 `face[a, b, c]`, `edge[a, b]`, `vertex[a]`, `outer[ ]` 형식으로 출력한다.

[입력] `points.txt` 파일의 첫 줄에 집합  $S$ 의 수  $N$ 이 주어진다. 그리고 이어지는  $N$ 개의 줄에 각 점  $s_i$ 의 좌표가 integer  $x_i, y_i$  으로 주어진다. 단  $1 \leq x_i, y_i \leq 1000$  이다. 그 다음 줄에는 질문점  $q_i$ 의 개수  $K$ 가 주어지고 이어서  $K$ 개 점의 좌표 그  $q_i=(x_i, y_i)$ 의 좌표를 나타내는 두 정수  $x_i, y_i$ 가 나온다. 출력은  $K$ 개 점에 대하여 각 점의 위치를 **face[a, b, c]**, **edge[a, b]**, **vertex[a]**, **outer[ ]** 형식으로 각 줄에 하나씩 출력한다. 아래와 같은 경우라면 4개의 질문점에 대한 출력은 다음과 같아야 한다. edge에 해당할 경우, 시작,끝 vertex는 오름차순으로 출력한다.



출력	
<code>outer [ ]</code>	// $q_1$
<code>face [1,4,9]</code>	// $q_2$
<code>outer [ ]</code>	
<code>vertex [3]</code>	// $q_4$
<code>edge [5,11]</code>	

[조건] `points.txt`의 점의 개수는 최대 100개 이하이다. 그리고 질문 점의 개수는 최대 20개이다. 제출할 프로그램은 `NAME_DTLocation.py`이다.  $DT(S)$ 와 질문점  $\{q_i\}$ 를 plot으로 그려서 답이 올바른지를 확인할 수 있도록 해야 한다. 자신이 사용한 데이터를 2종 이상 공개한다. 이것으로 다른 학생의 코드를 검사한다. `NAME_dt{01,02}.txt`로 한다. 마감은 8번 과제와 같은 시간이다.