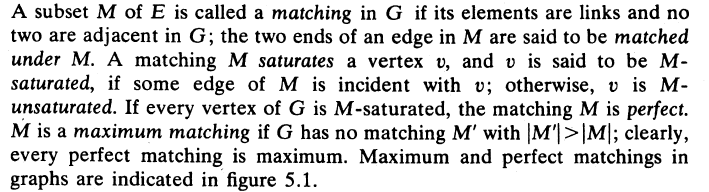
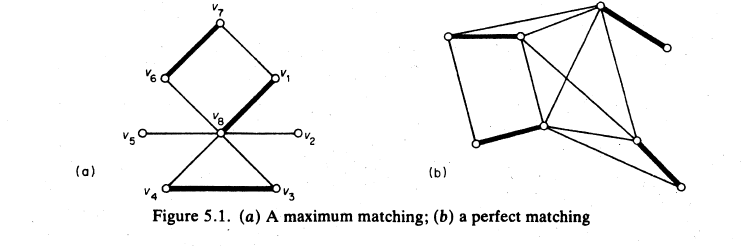
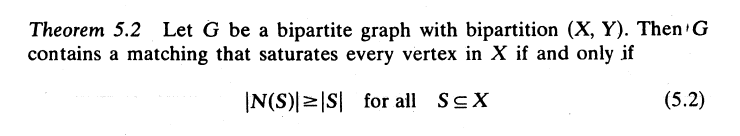
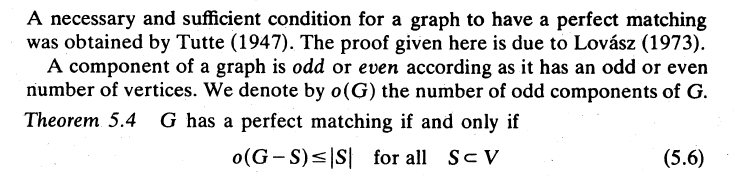
# 工程中的计算几何与图算法

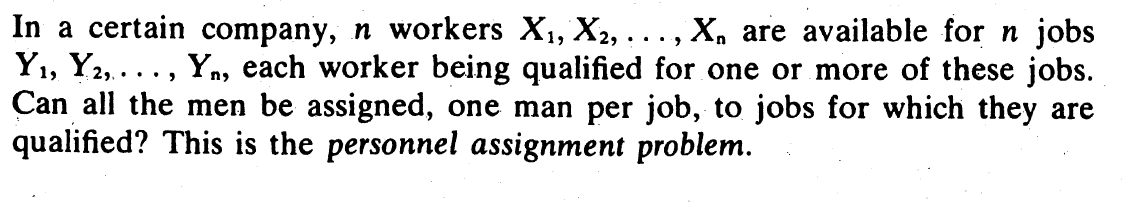
## 匹配性

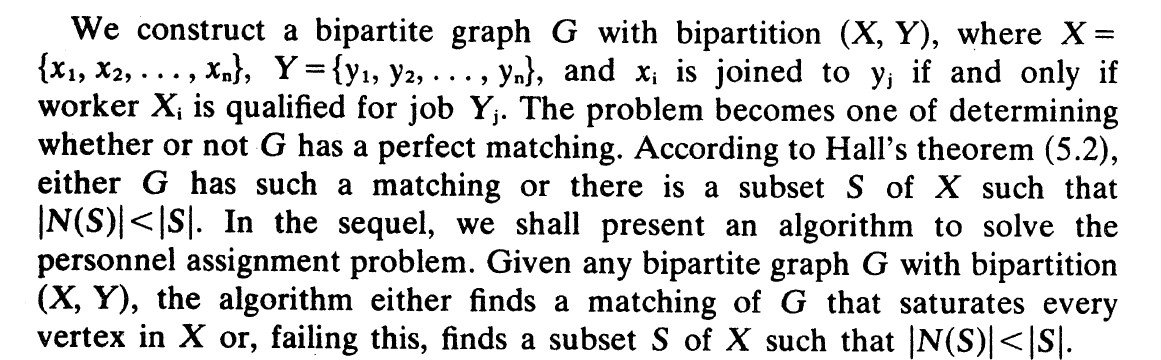








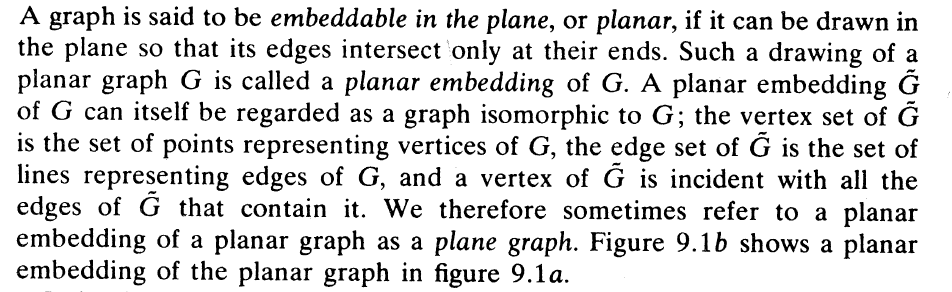


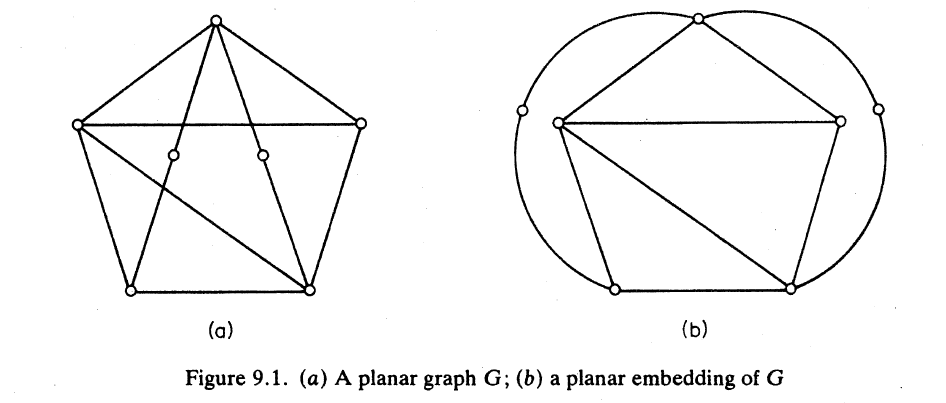


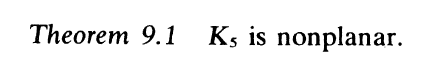
Algorithm：

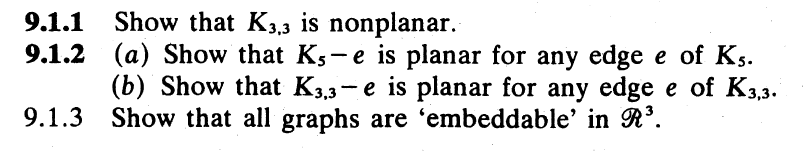


## 图的可平面性（桥）

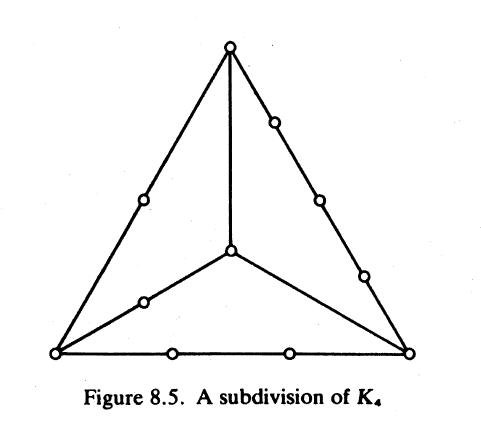
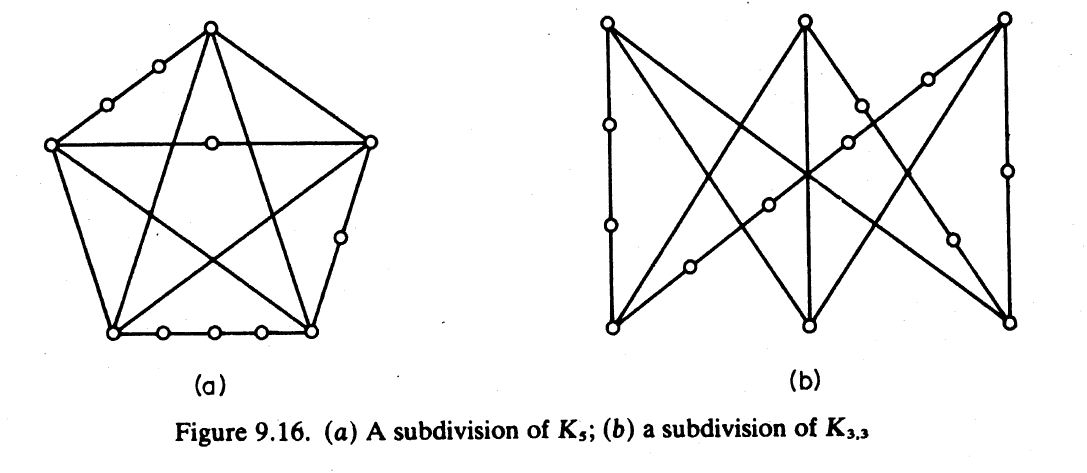


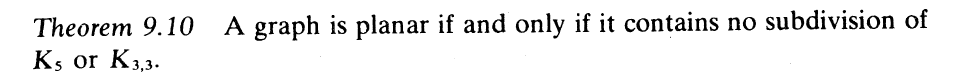




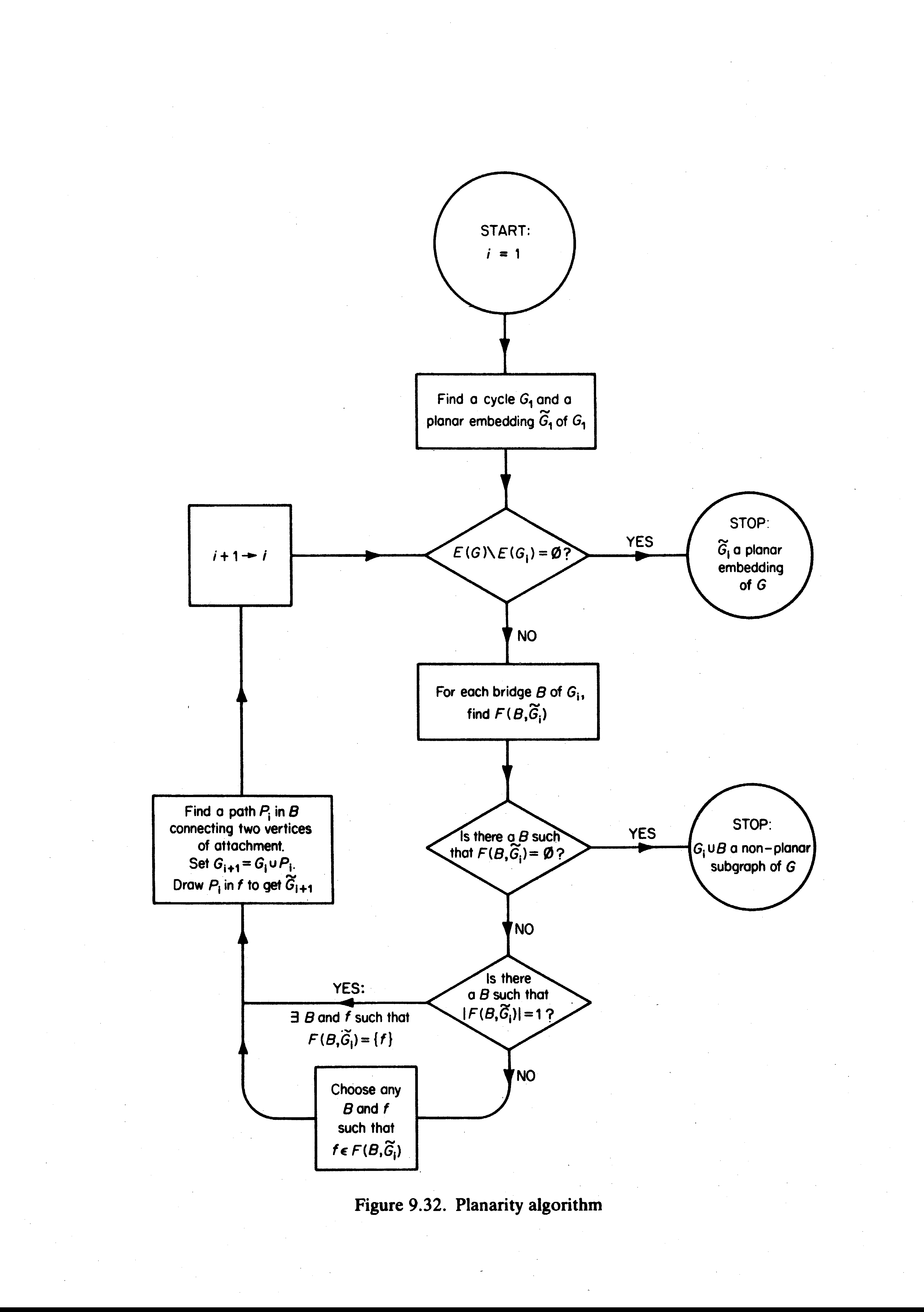




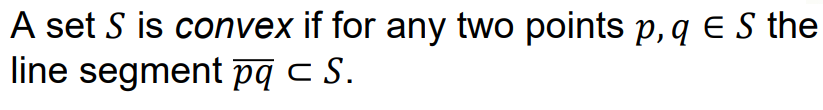


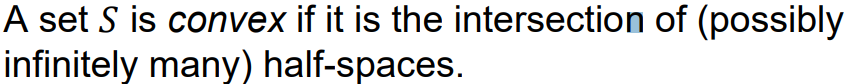


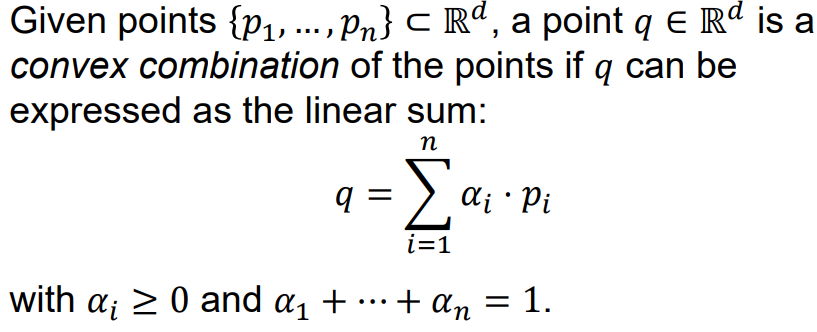


## 点集合凸包（Convex Hulls）

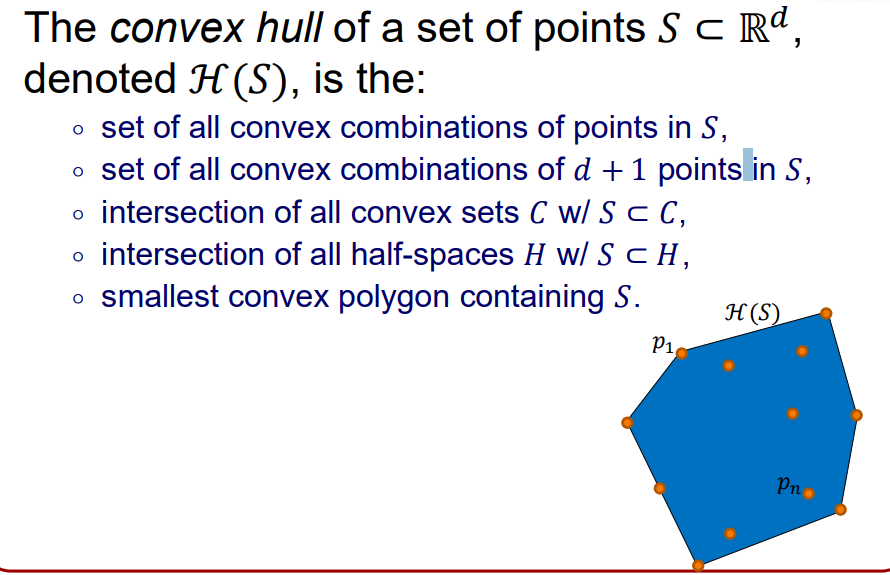
### Convexity

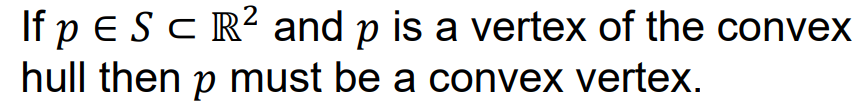


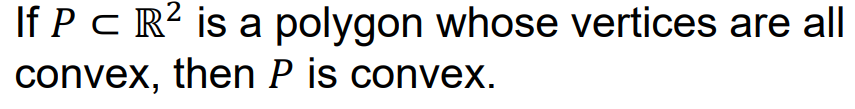


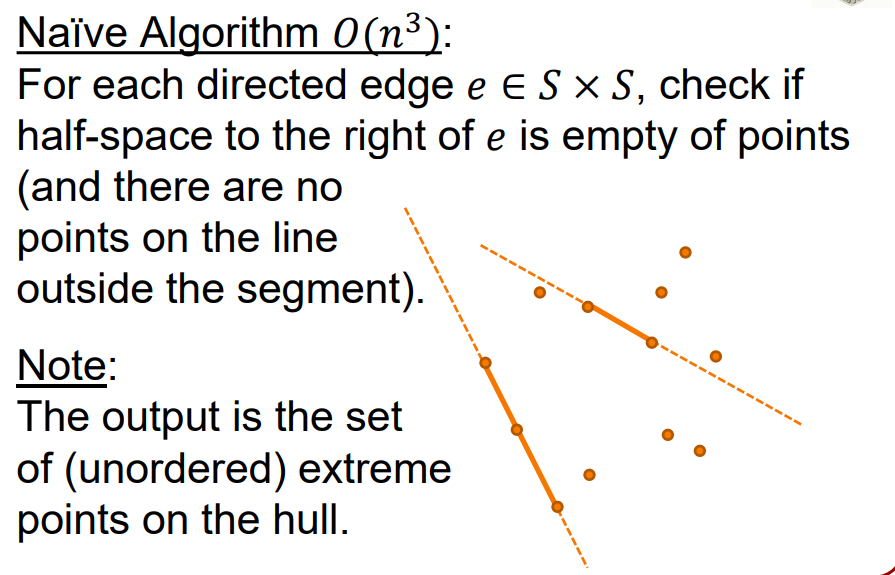


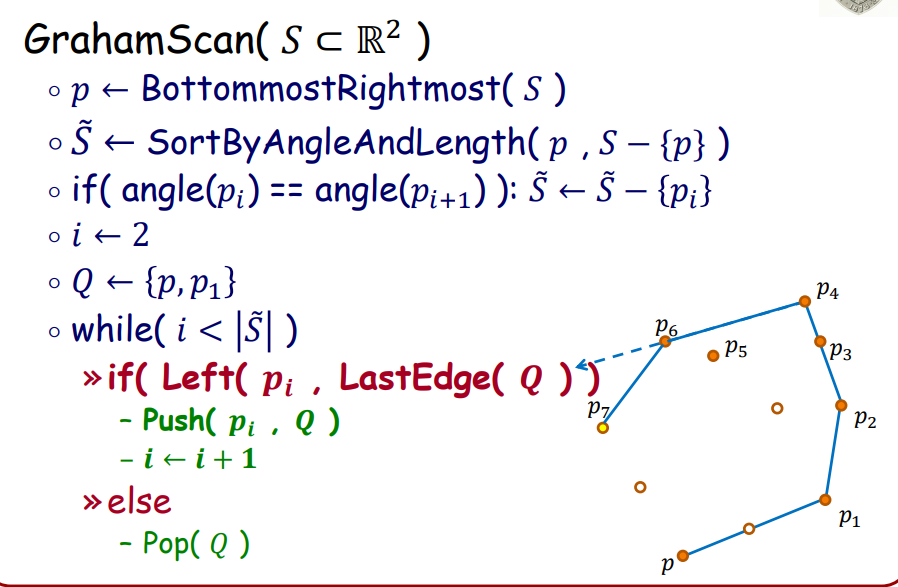
### Convex Hull

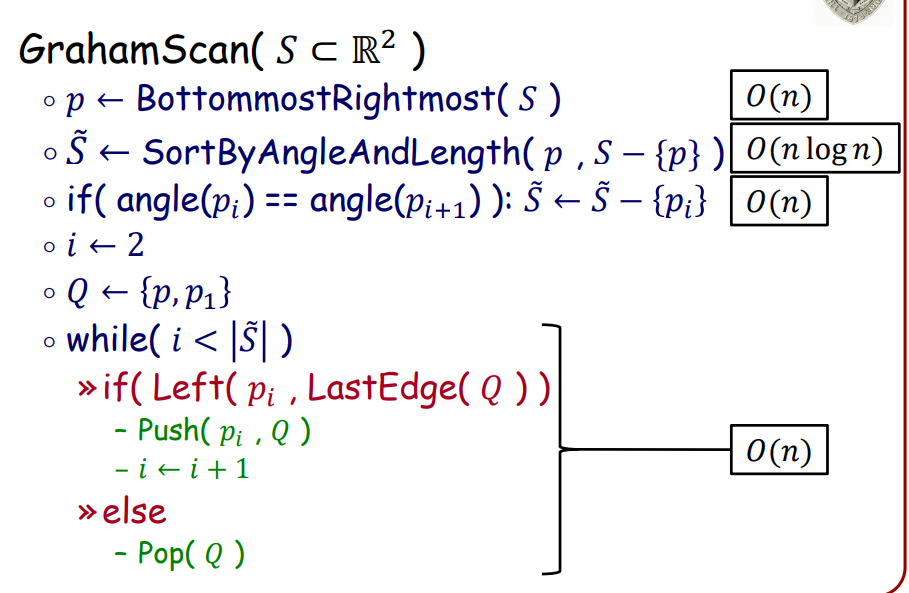


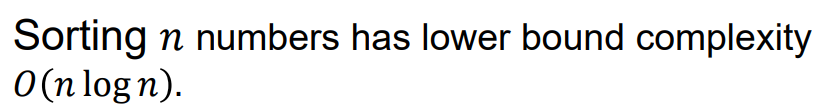












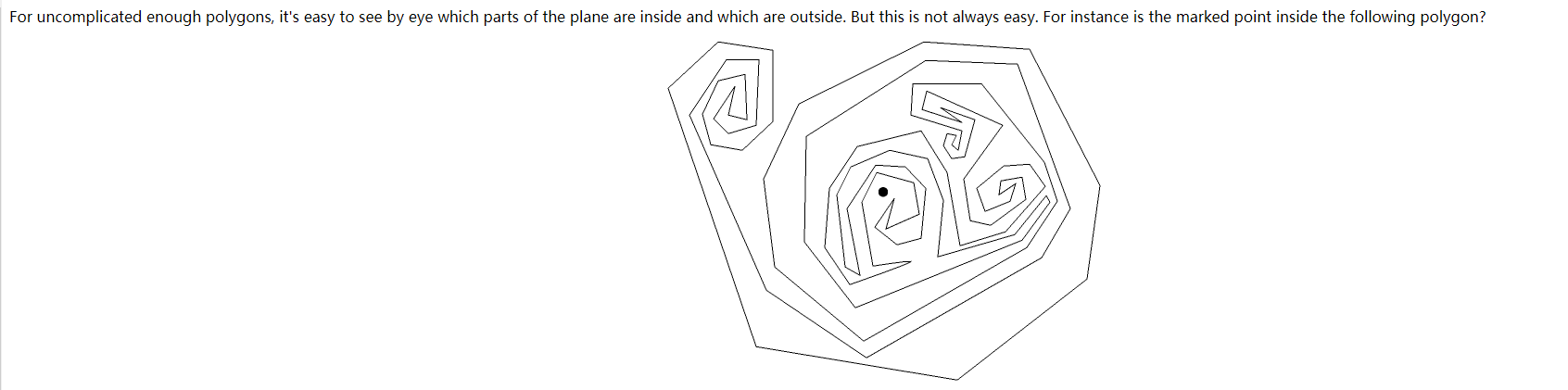
Tips：

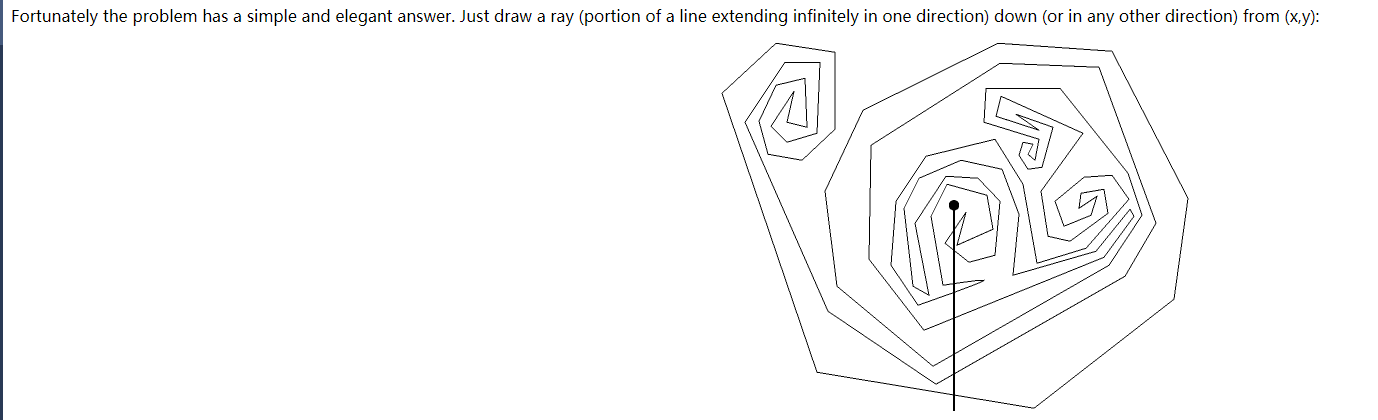
快速排序时间复杂度平均是O(nlgn)但最坏是O(n^2)

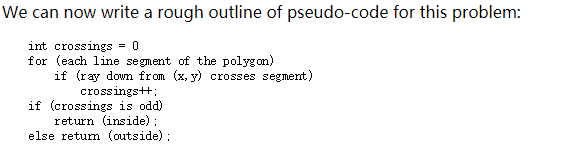
归并排序时间复杂度最好和最坏时间复杂度都是O(nlgn)

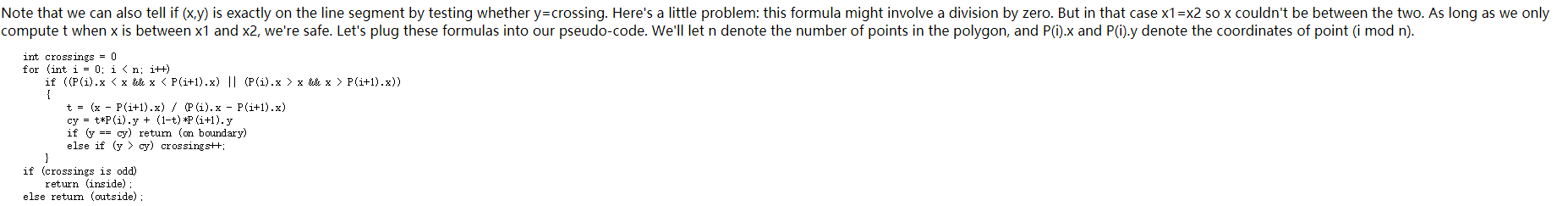
## 点与多边形的位置关系（Testing if a point is in a polygon）

Sometimes we use the phrase simple polygon to emphasize the requirement that no two segments cross.

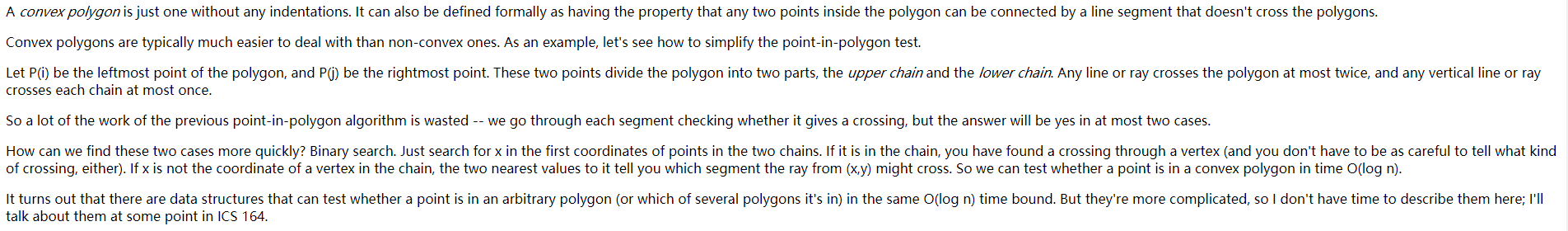


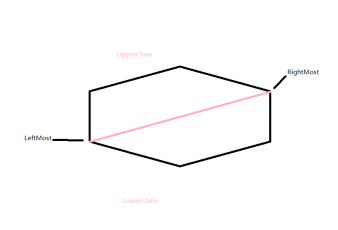




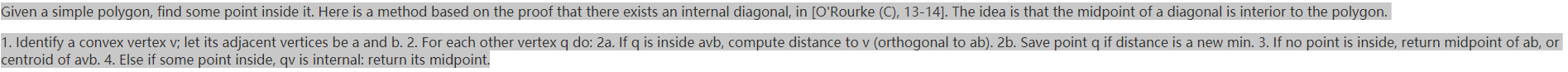


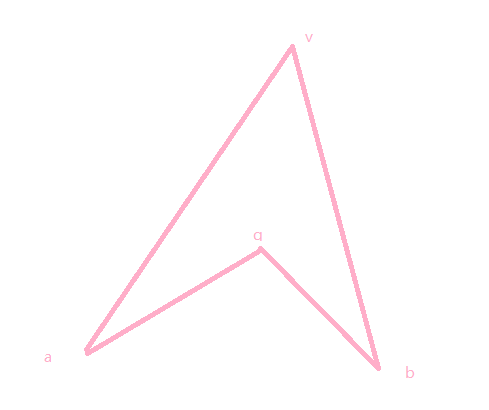
Points in convex polygons





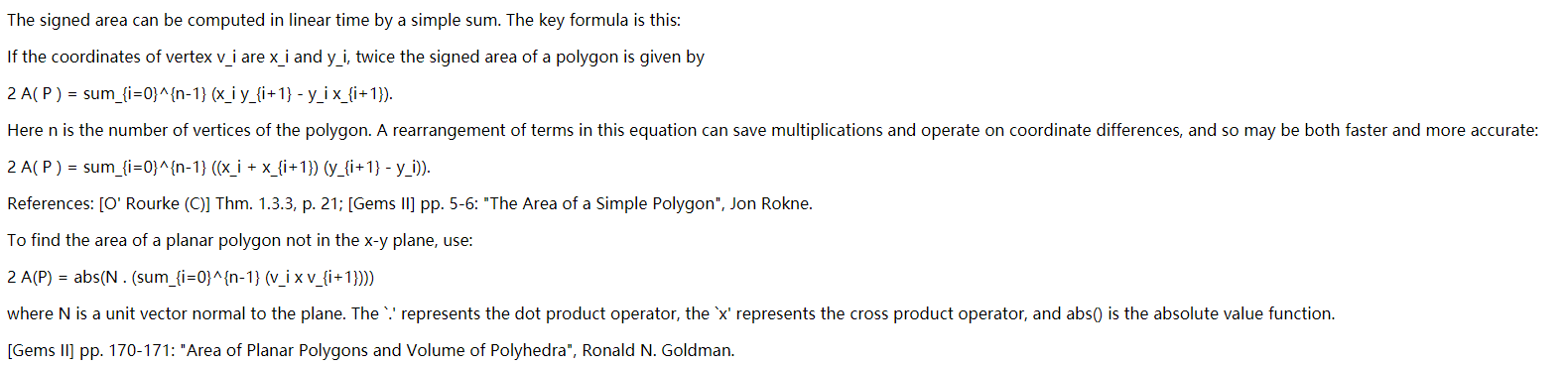
Finding a single point inside a simple polygon



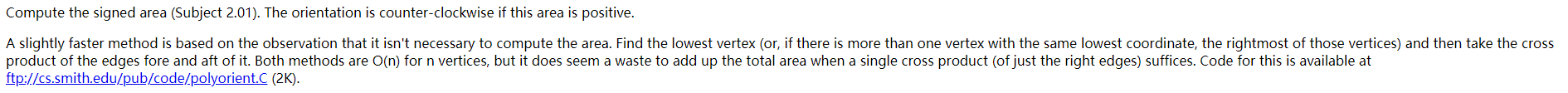


## 多边形面积（Area of polygon）及定向（orientation of polygon）

Area



Orientation



## 多边形凹凸性判定（Convex and Concave Polygons）

Every polygon is either convex or concave. The difference between convex and concave polygons lies in the measures of their angles. For a polygon to be convex, all of its interior angles must be less than 180 degrees. Otherwise, the polygon is concave. Another way to think of it is this: the diagonals of a convex polygon will all be in the interior of the polygon, whereas certain diagonals of a concave polygon will lie outside the polygon, on its exterior.

Cross product each vector one by one.

A polygon is a set of points in a list where the consecutive points form the boundary. It is much easier to figure out whether a polygon is convex or not (and you don't have to calculate any angles, either):

For each consecutive pair of edges of the polygon (each triplet of points), compute the z-component of the cross product of the vectors defined by the edges pointing towards the points in increasing order. Take the cross product of these vectors:

Given p[k], p[k+1], p[k+2] each with coordinates x, y:

dx1 = x[k+1]-x[k]

dy1 = y[k+1]-y[k]

dx2 = x[k+2]-x[k+1]

dy2 = y[k+2]-y[k+1]

zcrossproduct = dx1\*dy2 - dy1\*dx2

The polygon is convex if the z-components of the cross products are either all positive or all negative. Otherwise the polygon is nonconvex.

## 等边等角多边形（equilateral equiangular）

Polygons can also be classified as equilateral, equiangular, or both. Equilateral polygons have congruent sides, like a rhombus. Equiangular polygons have congruent interior angles, like a rectangle. When a polygon is both equilateral and equiangular, it is called a regular polygon. A square is an example of a regular polygon. The center of a regular polygon is the point from which all the vertices of the polygon are equidistant. Regular polygons have special properties that we'll explore in the next section. Below are some examples of equiangular, equilateral, and regular polygons.

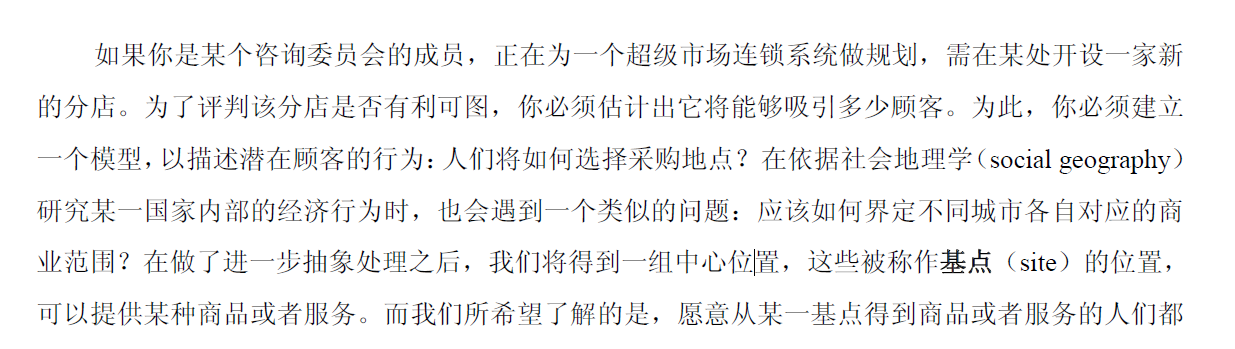


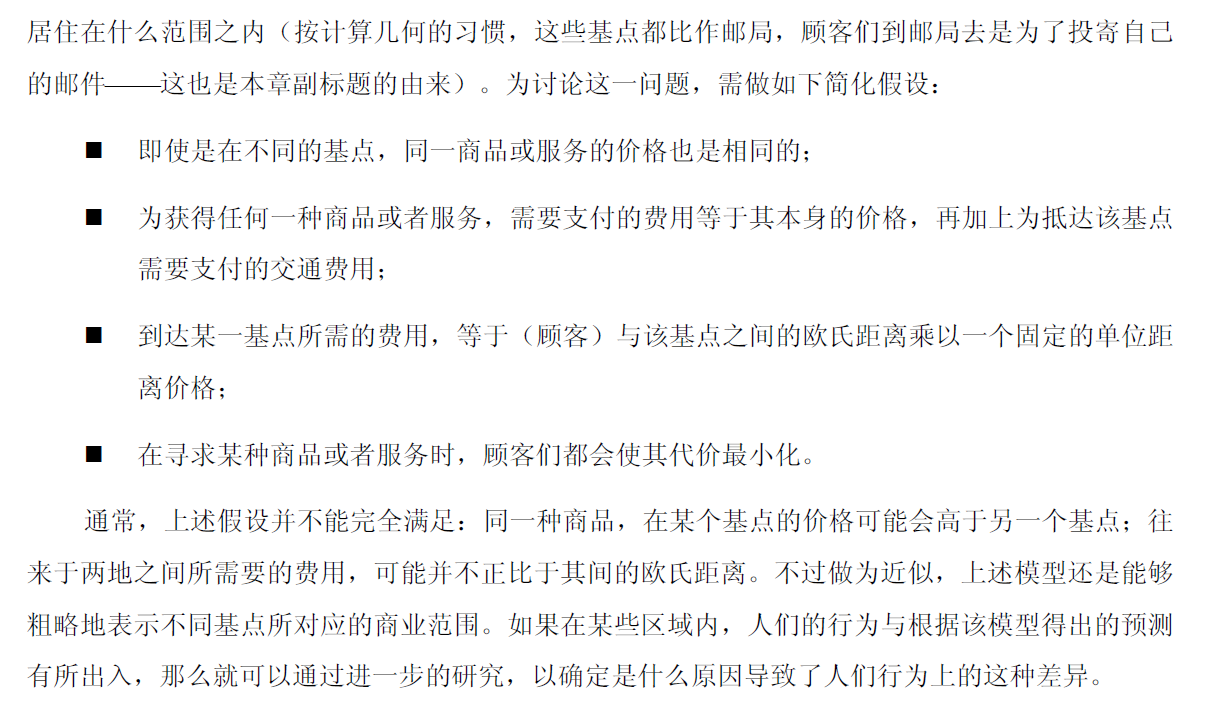
Q: Does regular polygon exist and only?

## 沃罗诺伊图（Voronoi Diagram）和德劳内三角剖分（Delaunay triangulation）

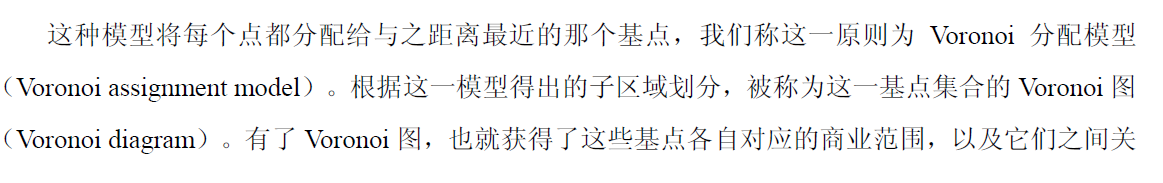
### Project Background

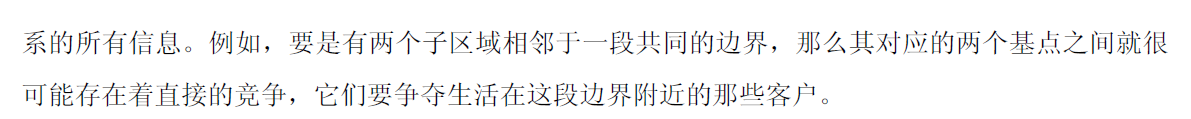
#### Voronoi Diagram

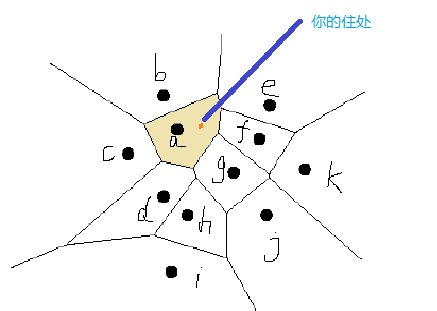




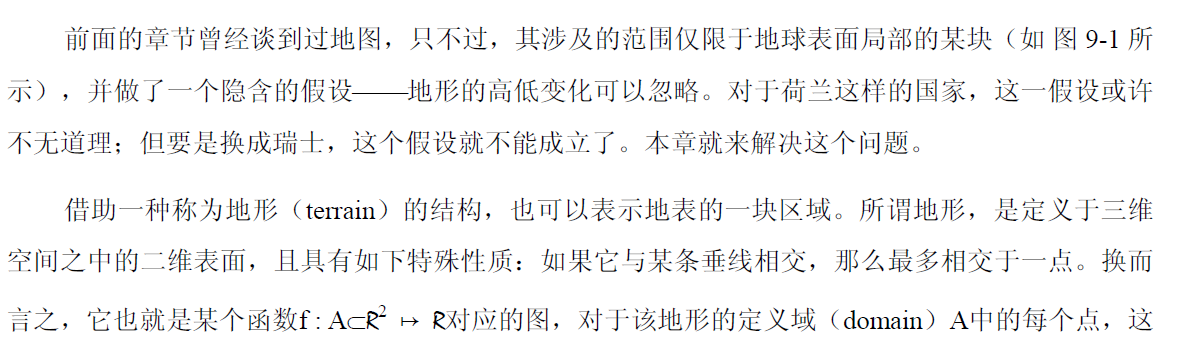


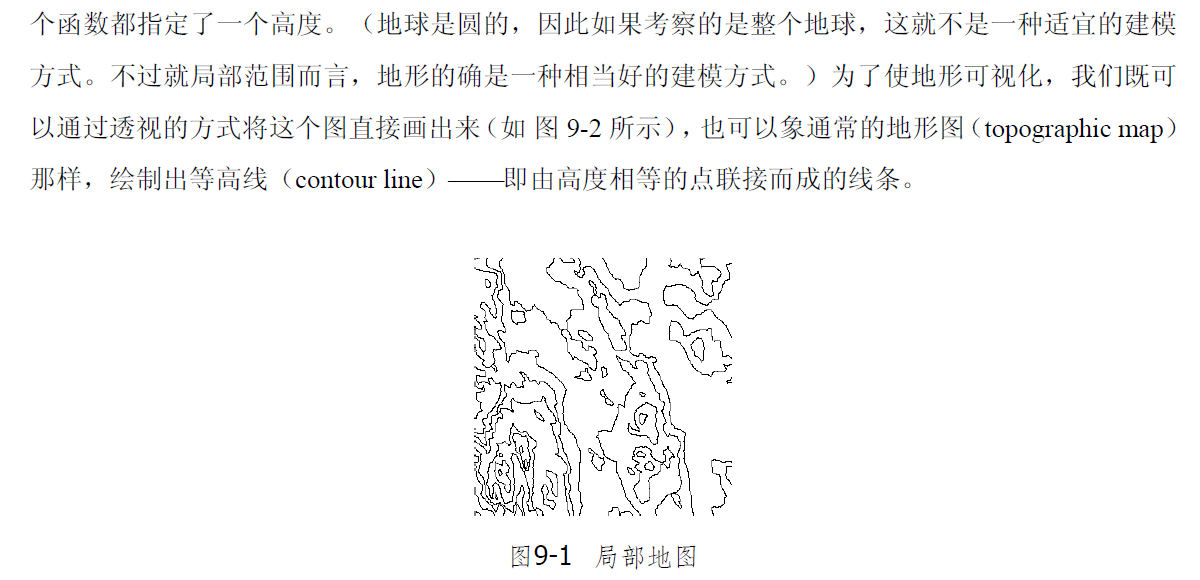


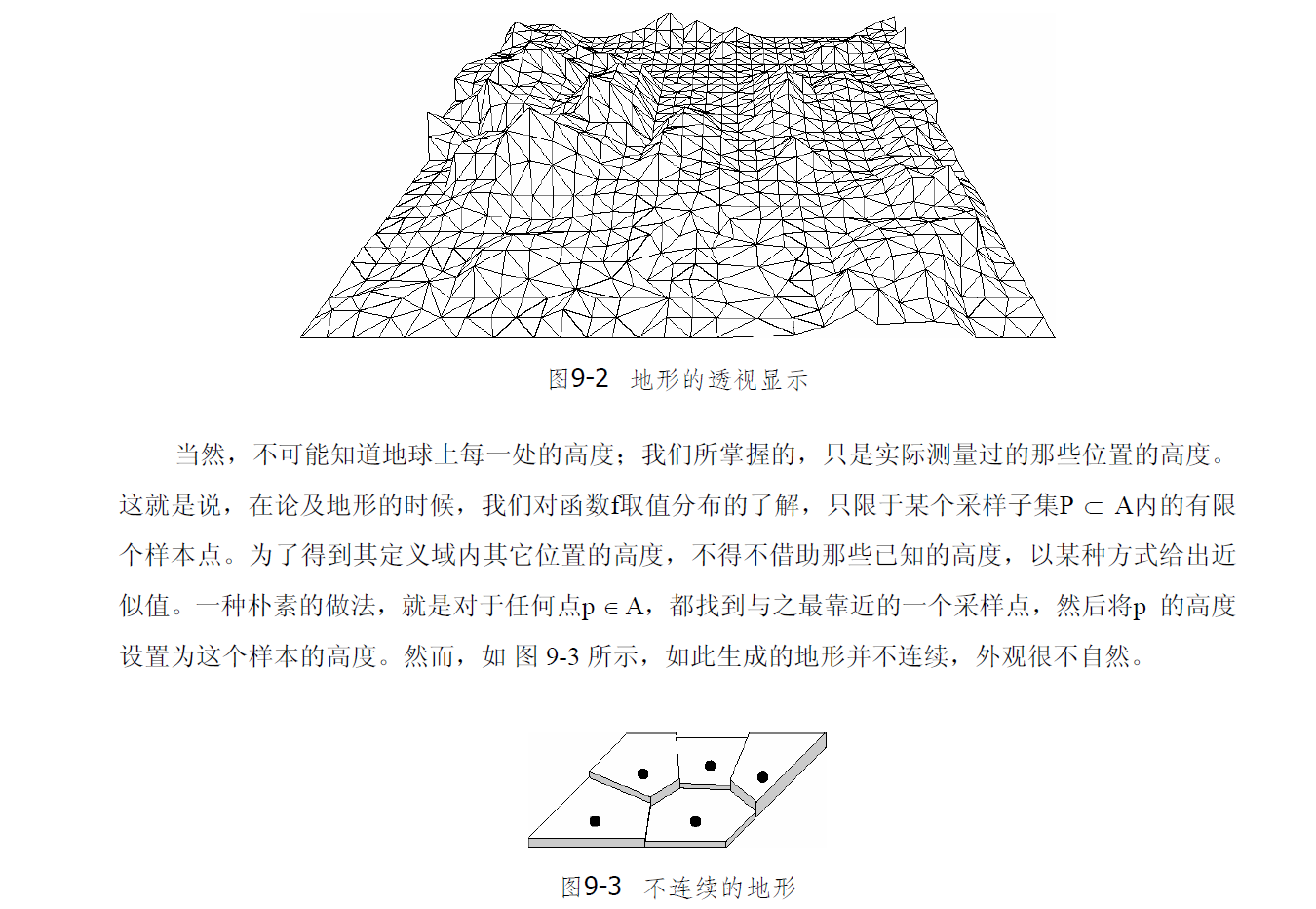


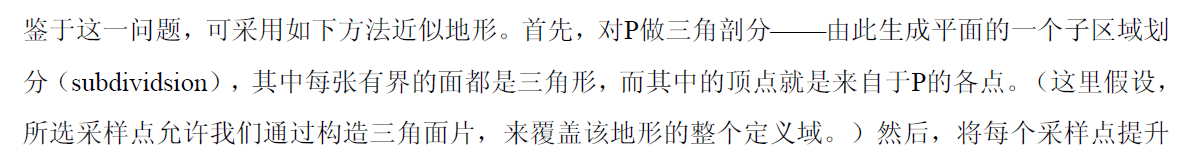


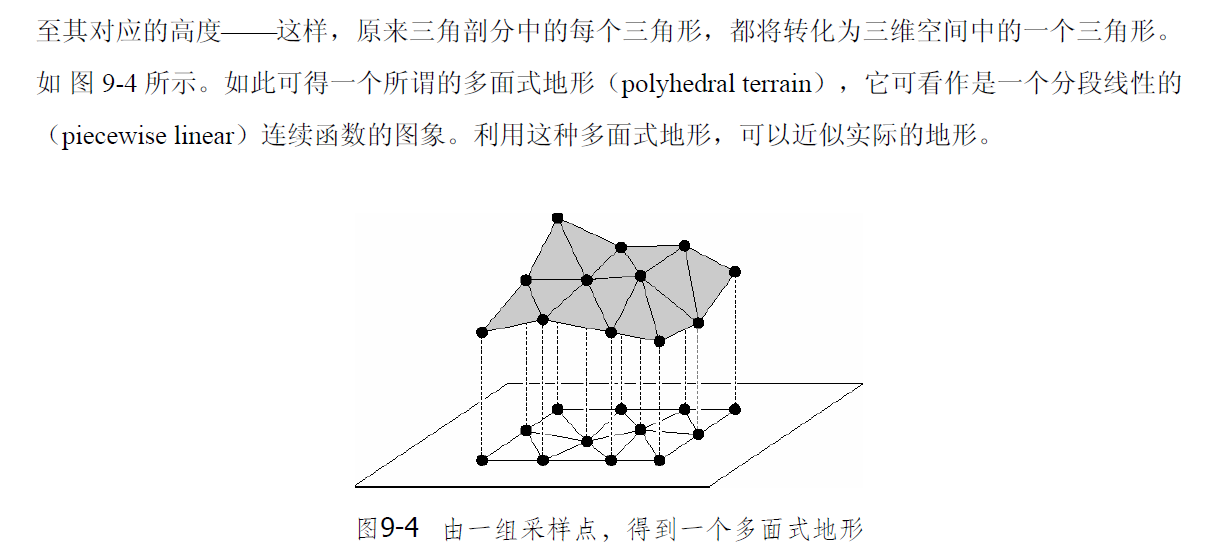
#### Delaunay triangulation

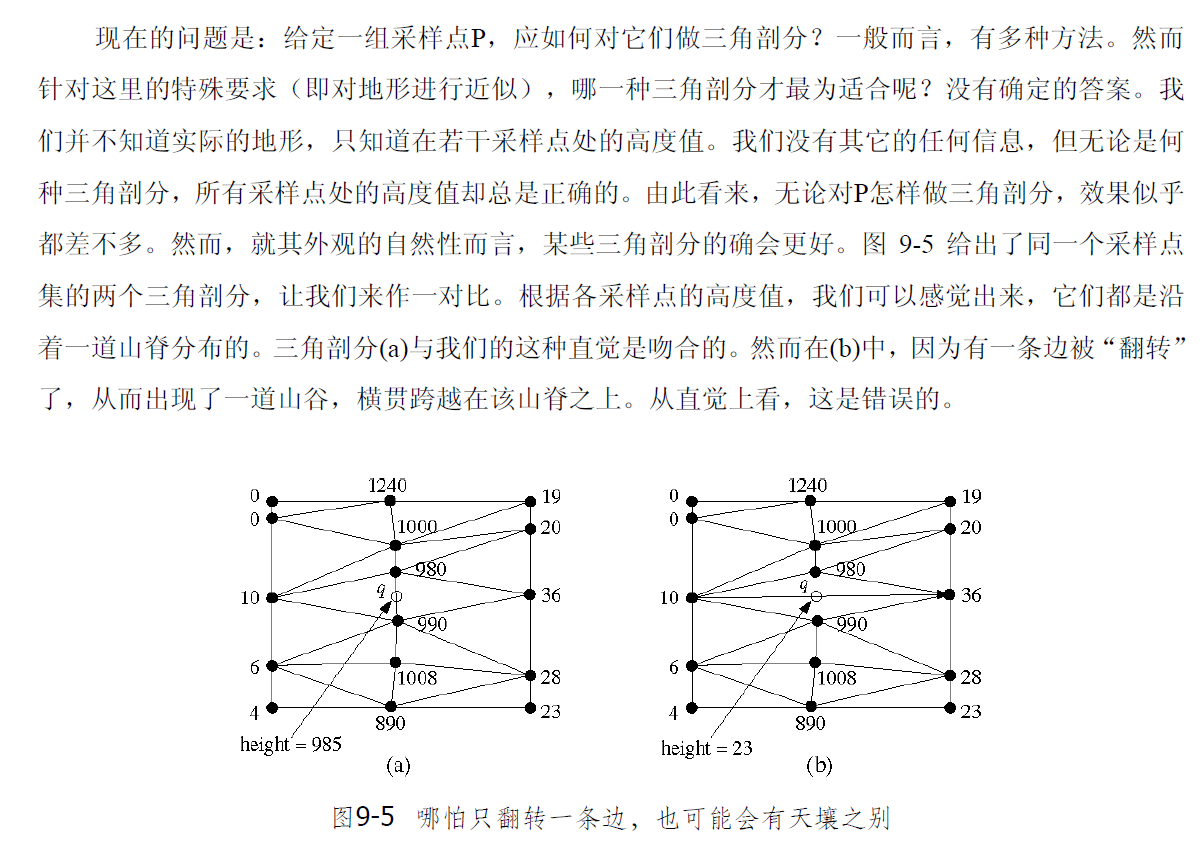


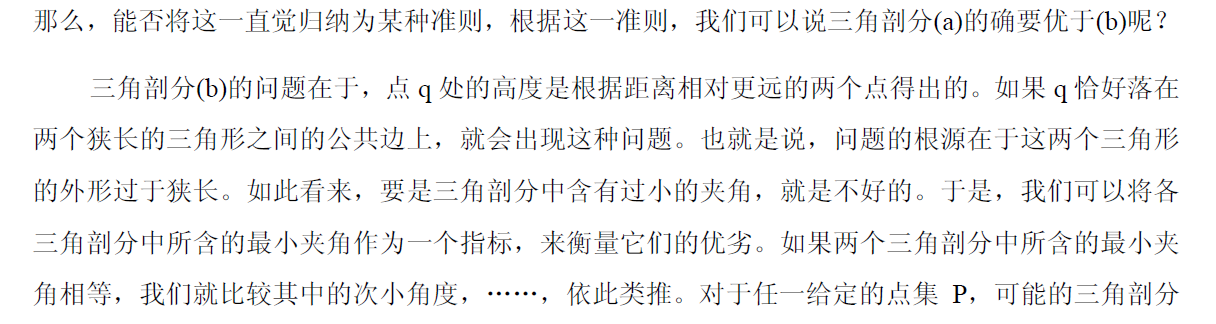


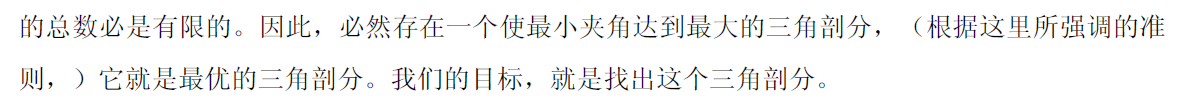




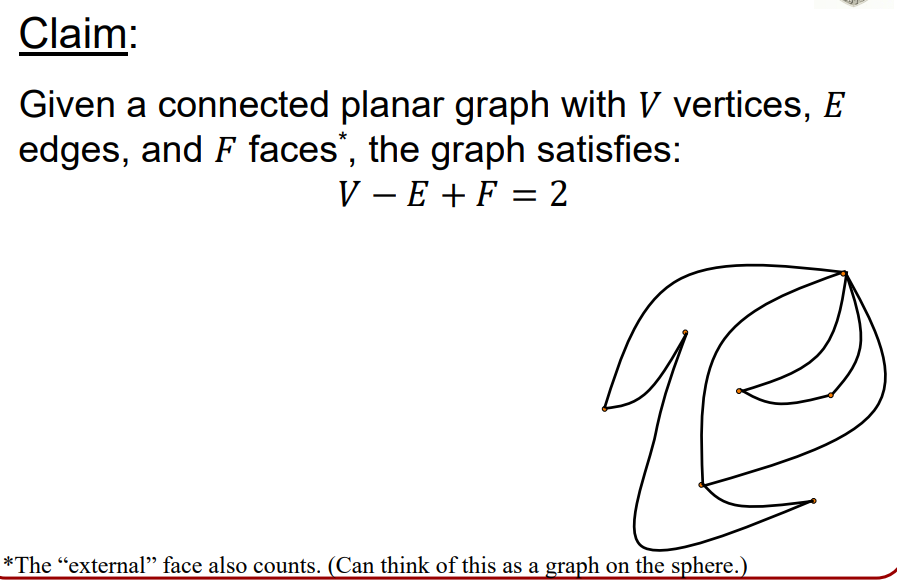


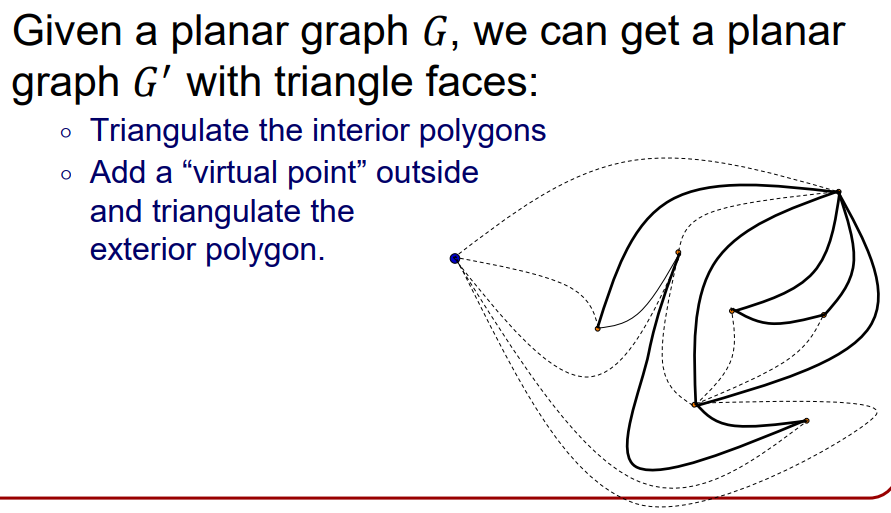


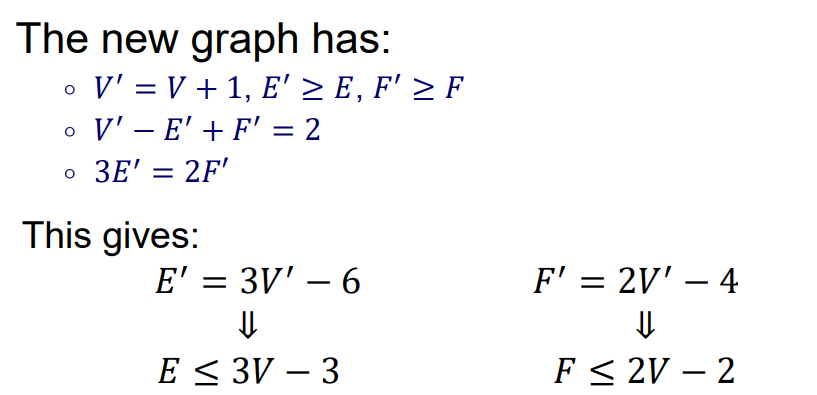


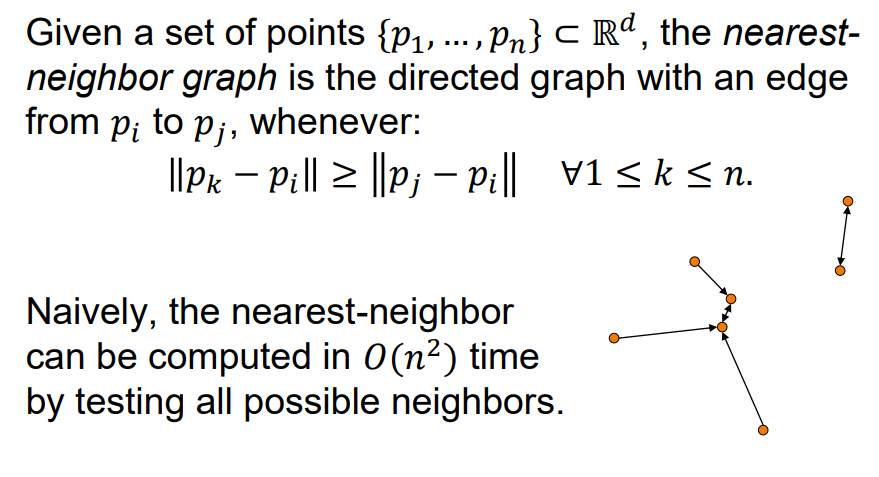


### Preliminaries

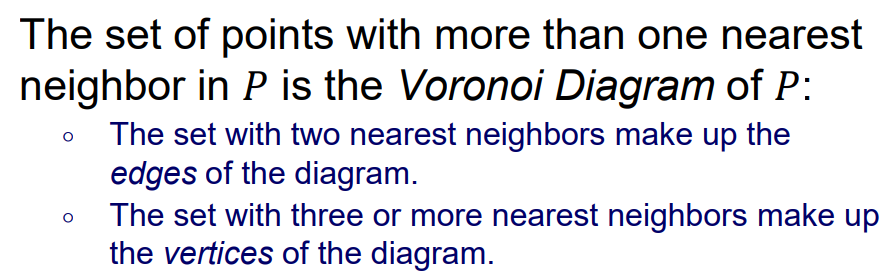




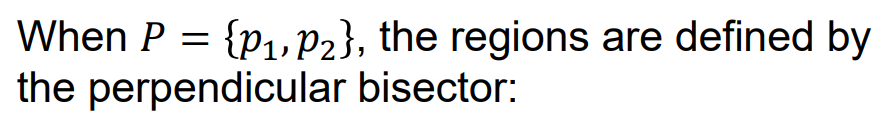


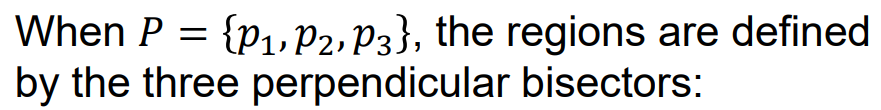


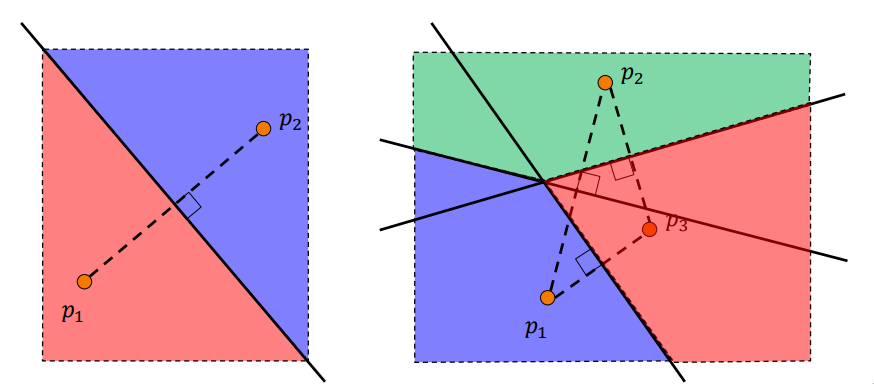
### Voronoi Diagrams

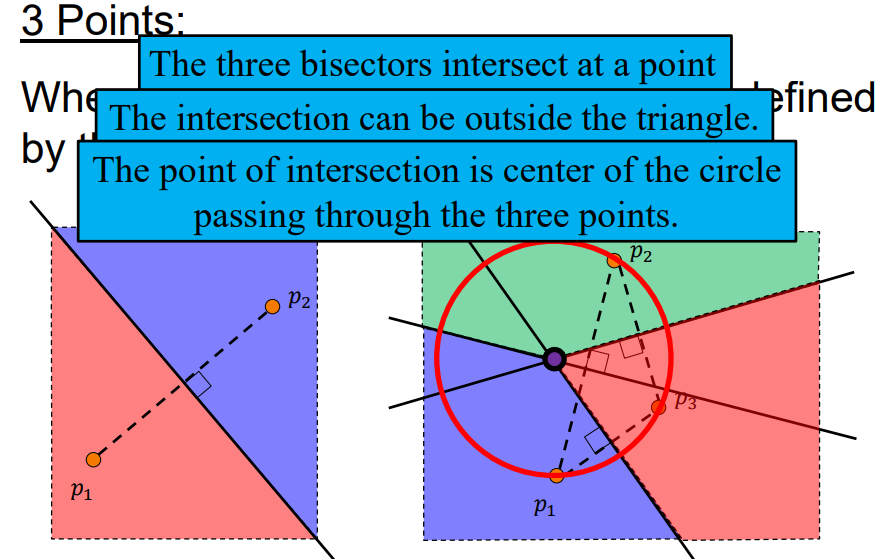


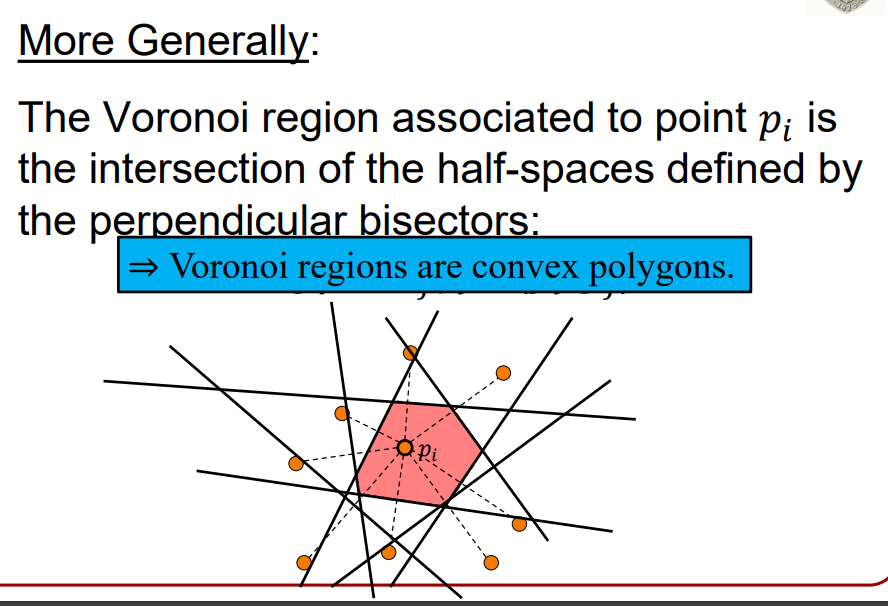




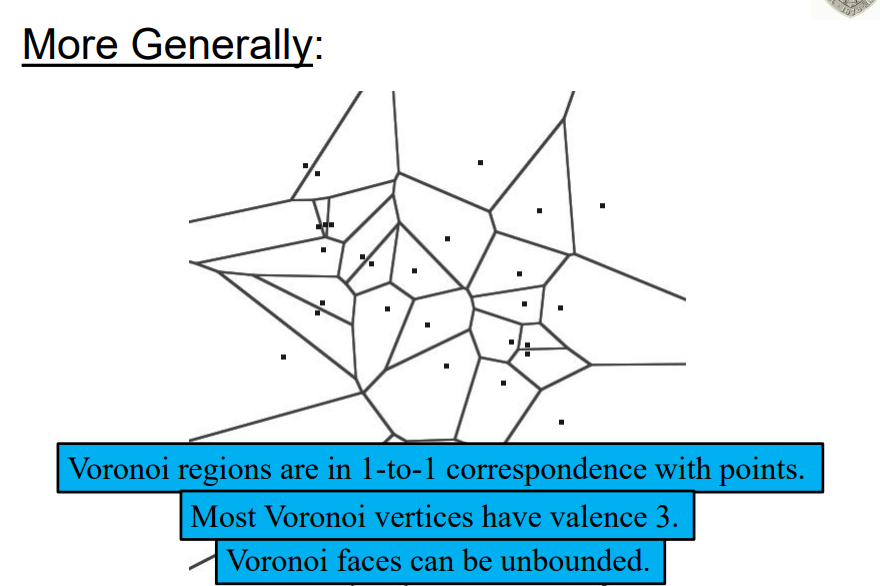


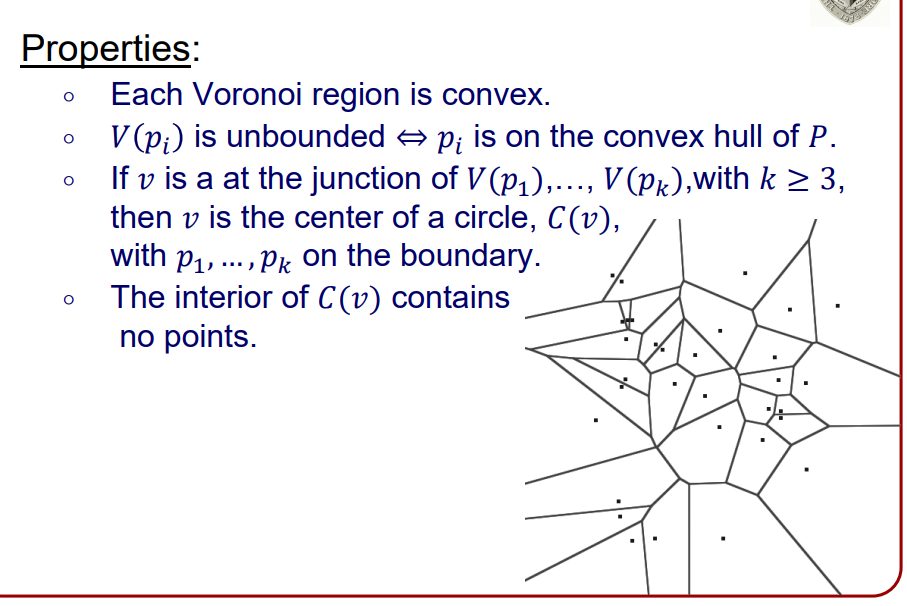




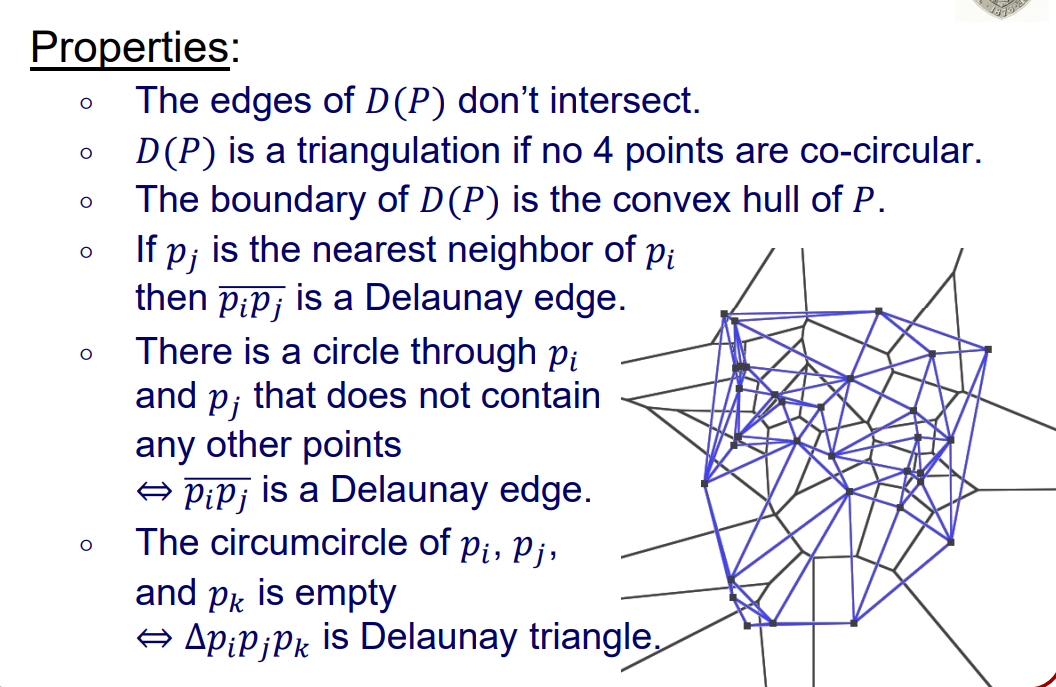


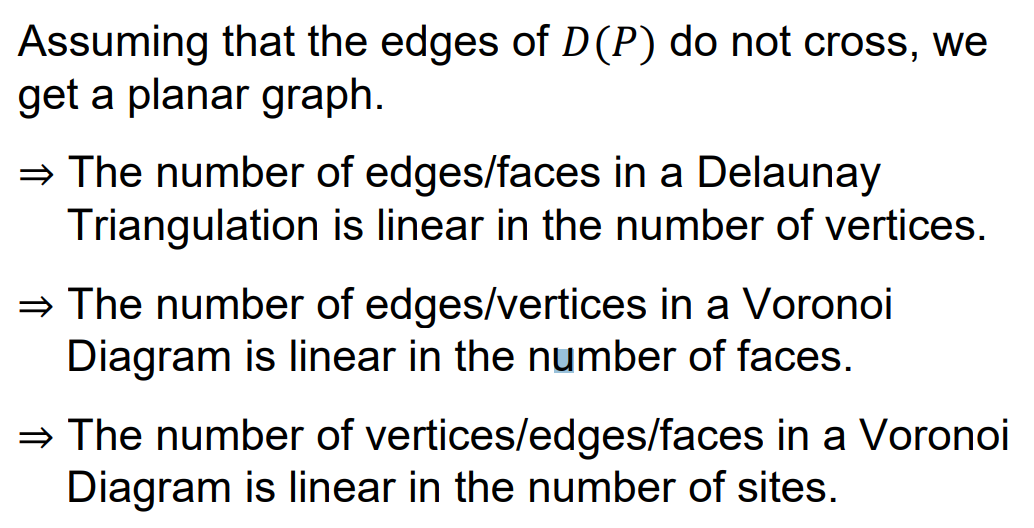


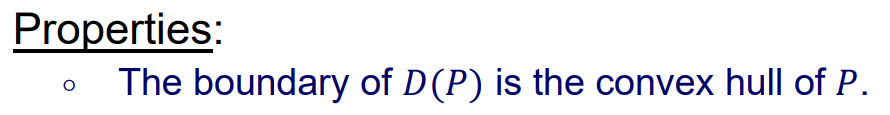




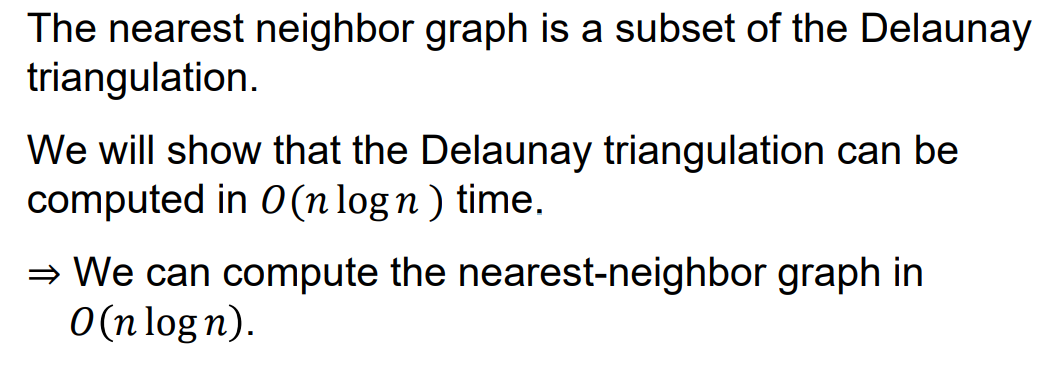
### Delaunay Triangulation

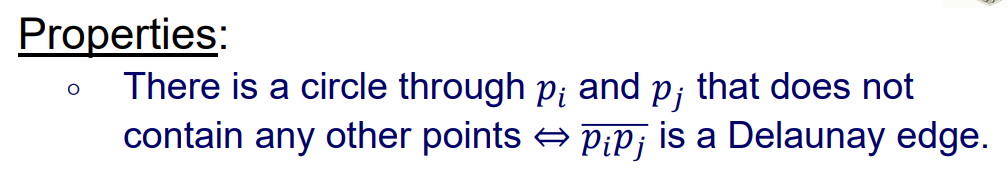


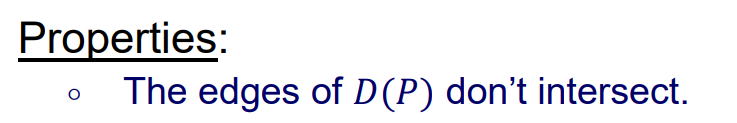






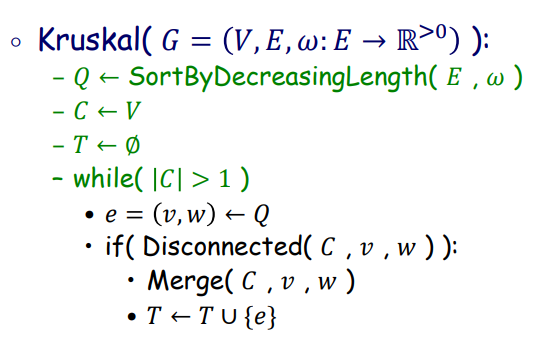




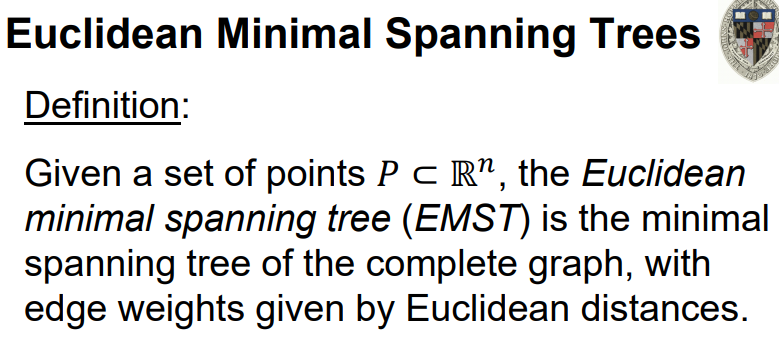


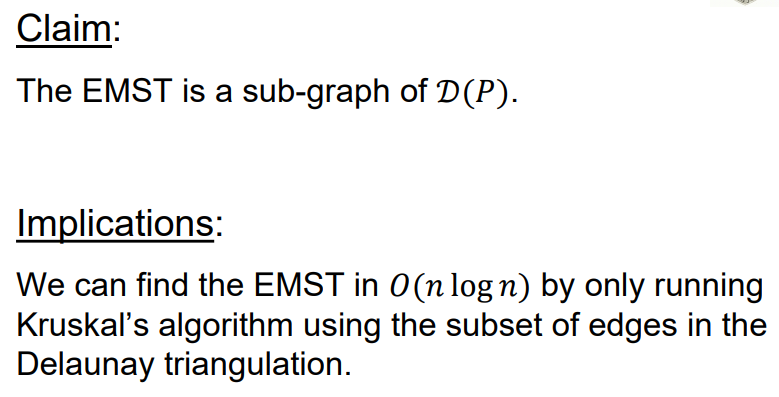
### Euclidean Minimal Spanning Trees



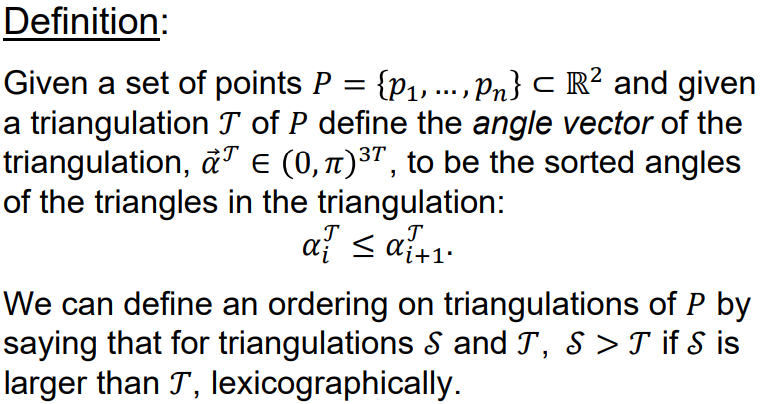


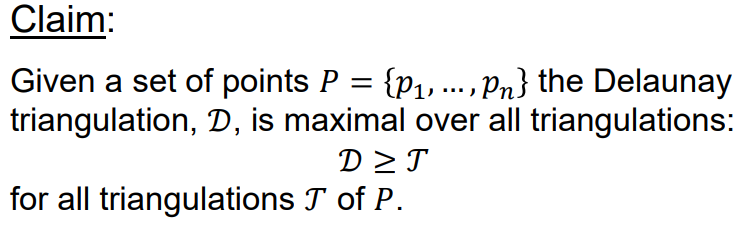




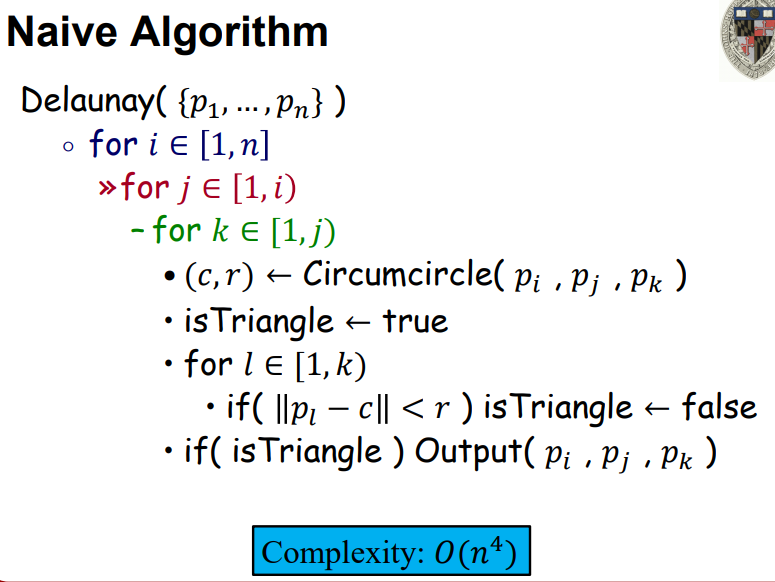


### Best Triangulation





### Algorithm



## 平面多边形区域划分（Polygon Partitioning）

## Arrangement2d