컴퓨터공학과 20211507 김동건

1) 임의의 개수의 포인트 셋에 대한 알고리즘을 수도 코드 형태로 작성 및 설명

임의의 개수의 포인트 셋에 대한 알고리즘 중, O(NlogN)의 시간복잡도를 갖는 incremental 알고리즘이 있다. 이는 모든 점을 포함하는 매우 큰 삼각형을 초기 삼각분할로 만들고, 점을 추가해가면서 진행되는 알고리즘이다. 점을 하나 추가할 때마다 그 점을 포함하는 기존의 삼각형들과 새로 추가한 점을 잇는다. 그리고, 추가된 세 삼각형을 각각 마주보는 삼각형과 들로네 삼각분할을 이루는지 판별한다. 만약 삼각분할을 이루지 않는다면, 공유하는 변을 지우고, 새로운 대각선을 긋는다. 이를 Flip이라고 한다. 이를 모든 점에 대해 반복한다. 수도코드는 다음과 같다.

Incremental\_Algorithm\_for\_Delaunay(points) :

Make Large Triangle that can contain all points  
 Add Large Triangle to triangulation  
 For [x, y] in points  
 Find triangle T that contains [x, y]  
 Subdivide T into three new triangle  
 For edge in NewEdges:  
 If not Satisfies Delaunay:  
 Flip(edge)  
 Remove Large Triangle  
 return triangulation  
2) Delaunay triangulation을 이용하여 Voronoi diagram으로 변환하는 알고리즘을 수도 코드로 작성 및 설명

먼저, 들로네 삼각분할에서 구한 각 삼각형들의 외심을 구한다. 그 후, 인접한 삼각형들의 외심을 서로 잇는다. 이를 통해 얻은 것이 Voronoi diagram이다.

Algorithm Delaunay\_To\_Voronoi(triangulation) :  
 voronoi\_node = empty set  
 voronoi\_edge = empty set  
 For triangle in triangulation :  
 get adjacent\_triangles  
 For T in adjacent\_triangles:   
 get circumcenter1 from triangle  
 get circumcenter2 from T  
 Insert edge(circumcenter1, circumcenter2)  
 Insert node(circumcenter1)  
 return voronoi\_node, voronoi\_edge