HomeWork 4 **Goldy**

I have used subdivision method to find a number of points lying on the Medial axis of the convex polygon.

Algorithm:

1. Generate a number of points on the edges of the polygon
2. Generate Delauney Triangulation for the points. Since most of the points are collinear, they must be handled carefully.
3. I have used the previous homework program for Delauney Triangulation. It was implemented by edge flipping algorithm. Vertices were inserted one by one in a large triangle – generating new triangles- "Legalizing" the surrounding edges recursively which is a local phenomenon.
4. When Triangulation is complete, find the circumcenter of the delauney faces of the polygon.
5. Add the distances between circumcenter of neighboring triangles to get an approximate length.
6. Since the circum-center would be lying on the Medial Axis the length found above gives an approximate length of the medial axis.

Main Challenges:

1. Dealing with collinear points. Due to floating point error points are not detected as collinear which results in formation of very small triangles. I have used a tolerance value for finding collinearity (cross product is zero). But the other problem is that some of the legal triangles are extremely small. So high tolerance means some of the triangle might be missed. So it may fail in some of the situation.

Time Complexity:

1. If each edge is subdivided into **'m'** pieces and there are **'n'** number of edges.
2. There are **O(nxm)** number of triangles.
3. For making a triangle delauney, its edges are flipped. But overall each edge is flipped only once.
4. Thus generating delauney triangulation is same as the order or number of triangles which is **O(nxm)**
5. For finding the triangle in which the point is to be inserted:
    a. The triangles are inserted in a tree data-structure with multiple connections among themselves (like a graph).
    b. A triangle is subdivided while inserting a point, the new triangles are children of it.
    c. When edges are flipped nodes can be cross linked as in a graph.
    d. So in case of a uniformily distributed point set the complexity of searching a triangle will be logarithmic but in worst in case it can be equal to the number of the triangles in the data-structure.

e. For **(mxn)** set of points the worst case complexity can be **O((mxn) (mxn))**. But Since in a convex polygon points are relatively distributed as they are chosen on a line. So we may assume its **O((mxn)log(mxn))**.

6. Calculation distance between point is **O(mxn).**

So the total complexity of finding medial axis assuming a uniform point set is **O((mxn)log(mxn))**. In super worst case it may be made **O((mxn) (mxn))**.