Name : Varre Harsha Vardhan

Entry_no: 2018mt60796

ASSIGNMENT-6

README

Introduction---

This assignment is based on graphs. The graph is implemented in order to store the 3d triangle. To make this a triangle graph the data-structures used are dynamic arrays and queues. Dynamic arrays are implemented using array. Queue is implemented with node as the use of queue requires no fixed length.

Queries and Complexities ----

There are four interfaces to implement. Firstly, point-interface it stores the coordinates of point. And all the triangles in which the point is presented.

Secondly, the edge-interface this stores the points of the edge and the points with which this is making a triangle.

Thirdly triangle-interface, this stores the points of the triangle. And also the adjacent triangles to the triangle.

ADD_TRIANGLE→

Here, I check the points given forms a valid triangle. And the points, edges and triangles are being stored in a dynamic array. If no of points are p, edges are e and triangles t. the order will be O(p^2+e^2+t^2). As I am checking the triangle is present or not.

MESH →

As the edges are having the points with which they are making triangles. I am checking the length of that point array to give the mesh type. The complexity is O(e).

BOUNDARY_EDGES→

This also can be done as the mesh so its complexity is also O(e);

COUNT_CONNECTED_COMPONENTS→

It is done by depth first search algorithm. We keep a Boolean in the triangle for it has been explored or not. It is done O(t^2). as it is travelled by two loops.

Neighbours of triangle →

It is done in O(t). there only one traversal in the list and comparing the points .

Edge neighbour of triangle→

It also in O(t). it takes O(t) to find the triangle.

Vertex neighbour of triangle→

This is also same as the edge neighbour.

INCIDENT_TRIANGLE→

It returns all the triangles of that point. This can be done in O(P). as finding the point takes order of p.

Neighbours of point →

The points that can form the edges with this point. this can be done in O(e).

Edge neighbour of point →

We should return the edges of that point. This is O(E).

Face neighbour of point →

We should return the triangle. This is O(T).

IS_CONNECTED→

Check the triangle is present or not if present explore it by dfs we get all the connected triangles. Check the second triangle is present or not.this O(T).

Triangle neighbour of edge→

All the triangles having this edge. It is done by O(e).

Centroid →

It take O(t^2). This done by DFS it should be done to find all components. Then give the centroid of those components.

Centroid of components →

This is same as centroid but to a fixed component.

Closest components→

It takes O(t^4). We should consider all the points and we should check for no edge in between and also it is not the same point.

Maximum_diameter →

We have to run breath first search with the help of queue. The max no. of hops is given as max diameter.it is O(t^2).