COMPUTATIONAL GEOMETRY
Assignment 2

Aman Choudhary
MTech Coursework, CSA 2020
Sr No: 17920

March 28, 2021

1 Problem 1

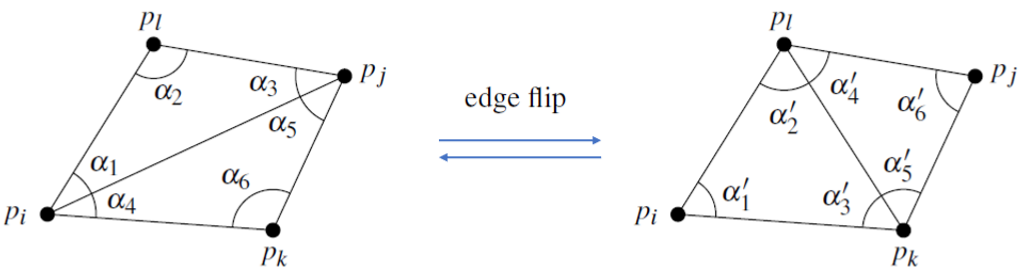
- Let us assume that we have a planar point set P. We can assume without any loss of generality that the point set P is **non-degenerate**, and hence that Delaunay Triangulation $\mathcal{T}_D$ corresponding to P is **unique**.

- We know that a triangulation $\mathcal{T}$ of P is **legal** if and only if $\mathcal{T}$ is a Delaunay triangulation of P. Since, we are assuming that P is non-degenerate, the only legal triangulation for P is its Delaunay triangulation, $\mathcal{T}_D$.

- We can compute a legal triangulation from an initial triangulation by simply flipping the illegal edges until all edges are legal. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be any two triangulations of the point set P. Let $a_1,\ a_2,\ \dots\ a_m$ be sequence of edge flips required to convert $\mathcal{T}_1$ to $\mathcal{T}_D$. This is a finite sequence, as the flipping algorithm is known to terminate. Similarly, let $b_1,\ b_2,\ \dots\ b_n$ be the sequence of edge flips required to convert $\mathcal{T}_2$ to $\mathcal{T}_D$.

- We also note that an edge flip is a reversible process, see below figure.

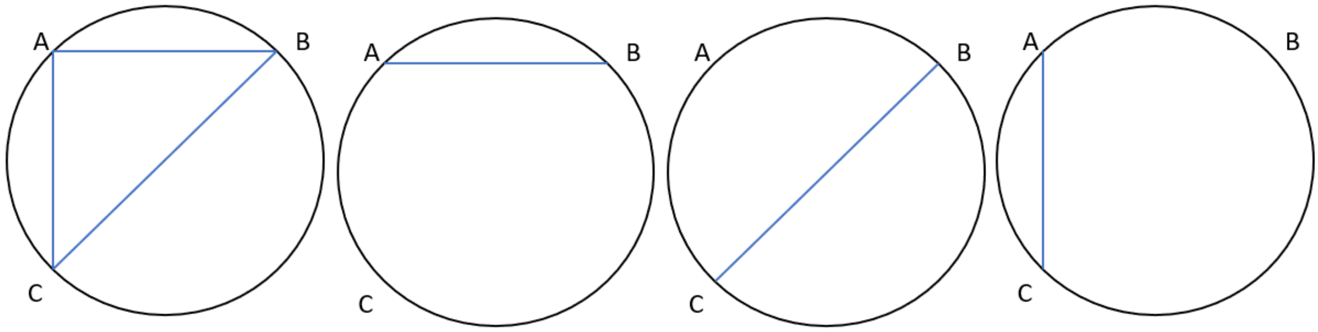
- Hence, we can convert $\mathcal{T}_1$ to $\mathcal{T}_D$ first using the sequence of edge flips namely: $a_1,\ a_2,\ \dots\ a_m$. Then we can convert $\mathcal{T}_D$ to $\mathcal{T}_2$ by reverting the sequence of changes followed in $b_1,\ b_2,\ \dots\ b_n$, namely $b'_n,\ b'_{n-1},\ \dots\ b'_2,\ b'_1$. We can follow a similar process while converting $\mathcal{T}_2$ to $\mathcal{T}_1$. Thus, we have shown that any two triangulations of a planar point set can be transformed into each other by edge flips.



2 Problem 2

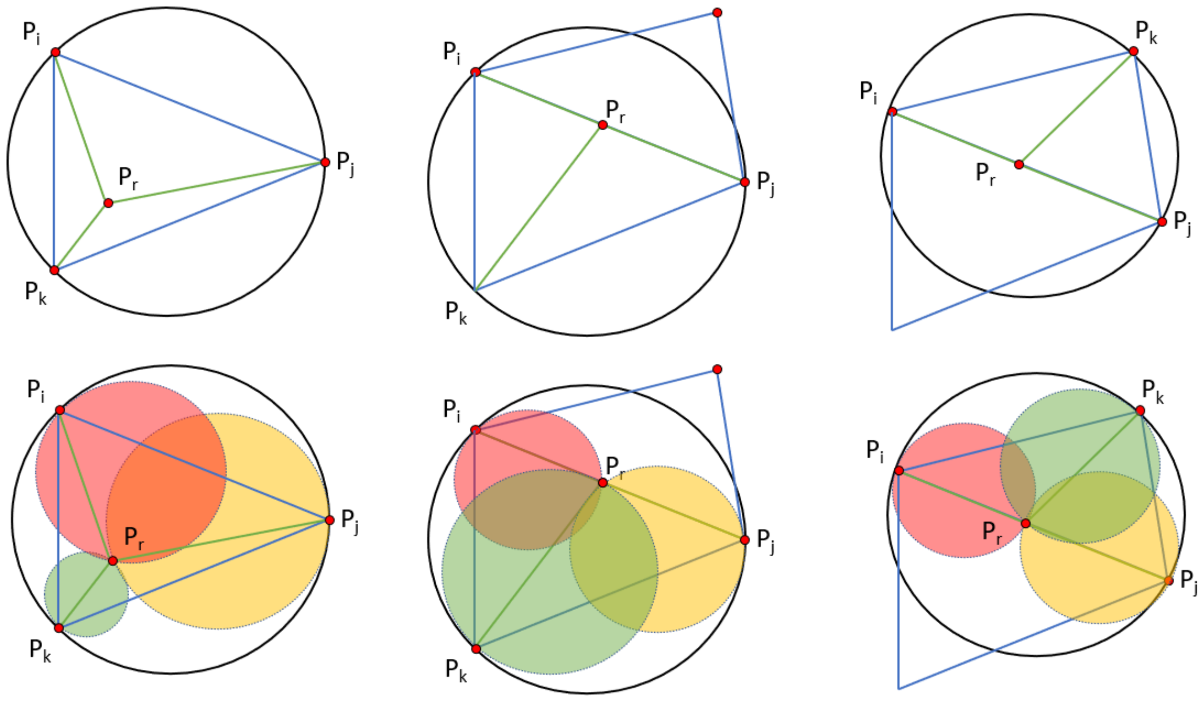
- **Triangle characterization for Delaunay triangulation**: Three points $p_i,\ p_j,\ p_k \in P$ are vertices of the same triangle of the Delaunay triangulation of P if and only if the circle through $p_i,\ p_j,\ p_k$ contains no point of P in its interior.
- **Edge characterization for Delaunay triangulation**: Two points $p_i,\ p_j \in P$ form an edge of the Delaunay triangulation of P if and only if there is a there exists an open disk C having p_i and p_j on its boundary which does not contain any other of P point in its interior.

Part A: Triangle characterization implies Edge characterization
Let us consider a triangle in the Delaunay triangulation, $\mathcal{T}_D$ of P, with vertices $A,\ B$ and C. Triangle characterization implies that the circle $\mathcal{C}$ circumscribing these points, will not contain any other point from P. As we can see from the below figure, this directly implies the edge characterization for the edges $AB,\ BC$ and CA if we take the same circle $\mathcal{C}$ as the empty disks for the edges. This can be extended to any general triangle following the same principle.



3 Problem 3

- During the incremental algorithm, when we add a new point P_r to an existing triangle with vertices $P_i,\ P_j$ and P_k, three new edges are created namely, $P_iP_r,\ P_jP_r$ and P_kP_r. Now, P_r could lie either in the interior of the triangle (see part (a) in figure below) or on some edge (see part (b) (c) in figure below). In either case, we can show that the newly formed edges will always be legal.
- Consider the circumcircle $\mathcal{C}$ for $P_i,\ P_j$ and P_k. Let us consider the point P_i and the centre O of the circle, $\mathcal{C}$. If, we now slowly shrink the circle, i.e. move the center O closer and closer to P_i along the radius OP_i, there will come a point when the boundary of the circle meets P_r. We stop at that point precisely to obtain a new circle $\mathcal{C}_i$ (shown in red below).
- We note that before adding P_r, the triangulation was legal, and hence the circle $\mathcal{C}$ circumscribing the triangle with vertices $P_i,\ P_j$ and P_k did not contain any other point from P. As a result when we shrink $\mathcal{C}$ to $\mathcal{C}_i$, we still have an empty disk which passes through P_i and P_r. Thus, edge characterization is satisfied for P_iP_r.
- In a similar, fashion, we can shrink the circle keeping P_j and P_k fixed in turns and obtain $\mathcal{C}_j$ (shown in yellow below) and $\mathcal{C}_k$ (shown in green below) respectively. Thus, we have shown that the edge characterization is satisfied for each of the newly formed edges and hence, all of them are legal edges.



(a) (b) (c)

4 Problem 4

Let P be a set of n points in general position, and let $q \notin P$ be a point inside the convex hull of P. Let p_i, p_j, p_k be the vertices of a triangle in the Delaunay triangulation of P that contains q. Now, q could lie either in the interior of the triangle (see part (a) in figure above) or on some edge (see part (b) (c) in figure above). In either case, we have shown above that the newly formed edges namely p_iq, p_jq and p_kq will always be legal. Since, adding q introduces only legal edges, the new edges belong to the Delaunay triangulation of $P \cup \{q\}$.

5 Problem 5

5.1 (a)

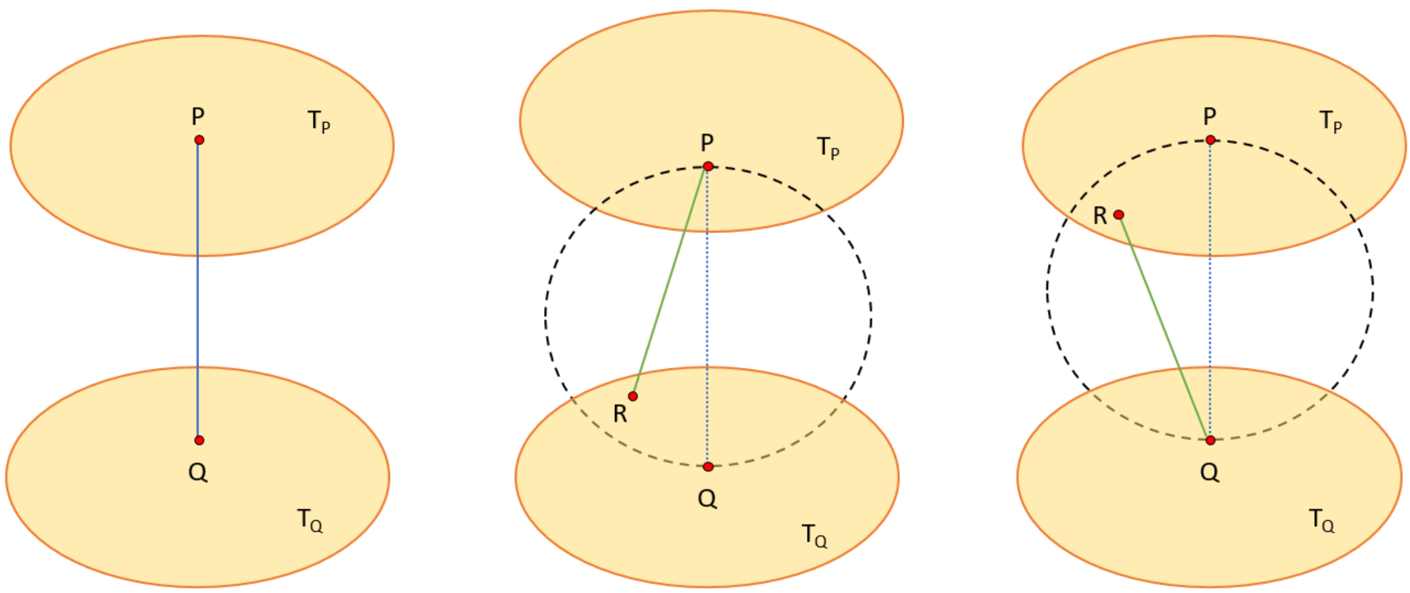
- Let T be the $EMST$ for the point-set $\mathcal{P}$. We assume that there exists an edge PQ in T, which does not exist in the Delaunay Triangulation, $\mathcal{T}_D$ of the point-set $\mathcal{P}$. Any edge in T is a bridge connecting two connected components. In this case, PQ connects T_P and T_Q.

- Since, we assumed that PQ is not in $\mathcal{T}_D$, there exists no open disk passing through P and Q which does not contain any other point from $\mathcal{P}$. Let us focus on one such circle $\mathcal{C}$ which has PQ as its diameter. It must contain at least one point $R \in \mathcal{P}$.

- Now, R can either belong to T_P or T_Q as illustrated in figure (b) and (c) respectively. Let us remove the edge PQ from the T to form $T' = T - \{PQ\} = T_P \cup T_Q$.

- We note that we can reconnect the two components by adding edge PR to T, hence obtaining $T'' = T' \cup \{PR\}$. The euclidean distance between P and R, i.e. $|PR| < |PQ|$ as any chord is shorter than the diameter of the circle.

- Now, $w(T'') = w(T) - |PQ| + |PR|$. Since, $|PR| < |PQ|$, we can say that $w(T'') < w(T)$. Thus, we have an $EMST$, T'' with a lower cost than T. Hence, we reach a contradiction that such a T exists in the first place. Therefore, any edge which belongs to the $EMST$ of a point set, must also belong to the Delaunay Triangulation of the point set.



(a) (b) (c)

5.2 (b)

We just proved that the $EMST$ for a point set is a sub-graph of its Delaunay Triangulation. We can compute the $EMST$ for $\mathcal{P}$ as follows:

- Compute the Delaunay Triangulation $\mathcal{T}_D$ for $\mathcal{P}$. The graph obtained contains $O(n)$ edges and vertices. This step takes $O(n \log n)$ time.

- Assign weights to edges in $\mathcal{T}_D$ using the Euclidean distance criteria. This step takes $O(n)$ time.

- Now, use Kruskal's algorithm on the resulting graph, $\mathcal{T}_D$ to compute the MST. The resulting MST is the required $EMST$. This step takes $O(n \log n)$ time.

The overall time complexity of the above algorithm is thus $O(n \log n)$.

6 Problem 6

6.1 (a)

Let us take any edge PQ, that exists in the Gabriel graph, $GG(\mathcal{P})$. For such an edge, the disc with diameter PQ does not contain any other point of $\mathcal{P}$. Thus, the edge PQ satisfies the edge characterization property for Delaunay Triangulation. Hence, it lies in the Delaunay Triangulation of $\mathcal{P}$. As a result, we can say that if an edge belongs to $GG(\mathcal{P})$, it will definitely belong to its Delaunay Triangulation as well. Hence, we can conclude that the Delaunay graph contains the Gabriel graph of point set $\mathcal{P}$.

6.2 (c)

We know that Gabriel Graph of a point set $\mathcal{P}$ is a sub-graph of the Delaunay Triangulation $\mathcal{T}_D$ for $\mathcal{P}$. It can be computed as follows:

- Compute the Delaunay Triangulation $\mathcal{T}_D$ for $\mathcal{P}$. The graph obtained contains $O(n)$ edges and vertices. This step takes $O(n \log n)$ time.
- For every edge PQ in $\mathcal{T}_D$, we need to check if the disc with diameter PQ contains any other point of $\mathcal{P}$ or not. If it does, then that edge cannot belong to Gabriel Graph and hence it is deleted. Here, we only need to check the points which are neighbours of P and Q. For every vertex u, there are $degree(u)$ neighbouring vertices. If we sum all the degrees it is equal to twice the number of edges which is $O(n)$. Hence, we need to perform $O(n)$ tests, each of which can be done in constant time. Hence, this step takes $O(n)$ time.

When the algorithm terminates, we are left with a Gabriel Graph since all the edges which do not satisfy the required property are deleted. Also, the overall time complexity of the algorithm is $O(n \log n) + O(n) = O(n \log n)$.

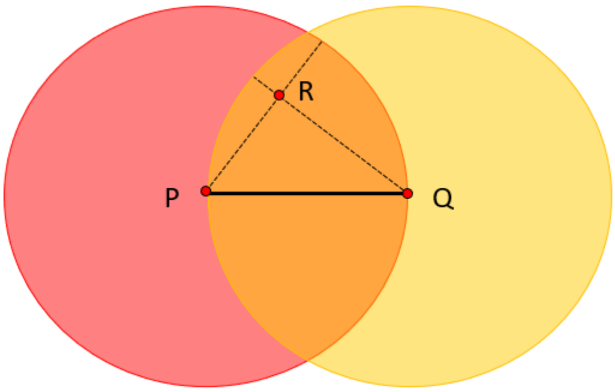
7 Problem 7

7.1 (a)

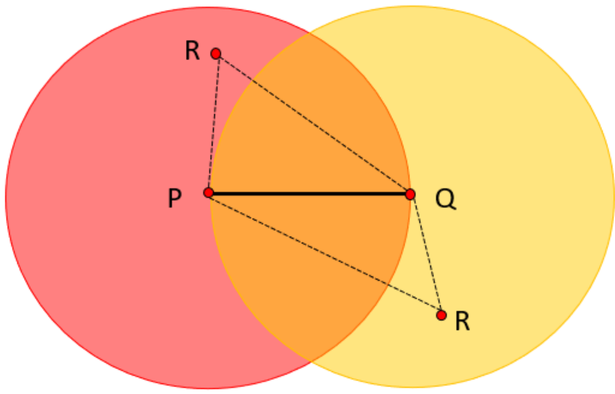
Part (a): If p and q are connected in the relative neighborhood graph (RNG), then $lune(p, q)$ does not contain any point of P in its interior.
Let us assume that p and q are connected in the RNG but $lune(p, q)$ does contain a point of R in its interior. From figure (a), we can see that $|PR| < d(p, q)$ and $|QR| < d(p, q)$, where $d(p, q)$ is the radius of the two circles. However, this implies that $max(|PR|, |QR|) < d(p, q)$. This contradicts the fact that PQ should be connected in the RNG.

Part (b): If $lune(p, q)$ does not contain any point of P in its interior, then p and q are connected in RNG
Let us assume that $lune(p, q)$ does not contain any point of P in its interior. Any neighbour point R must be outside the $lune(p, q)$ region as shown in figure (b). For any such R, $d(p, q) \leq max(|PR|, |QR|)$. Hence, there must be an edge between p and q in the RNG.

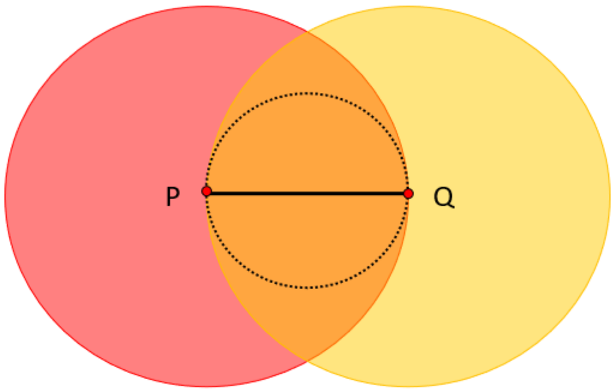
From **Part(a)** and **Part(b)**, the we conclude that p and q are connected in the relative neighborhood graph if and only if $lune(p, q)$ does not contain any point of P in its interior.



(a)



(b)



(c)

7.2 (b)

Let us take any edge that exists in the RNG of P. For such an edge, the region $lune(p, q)$ would not contain any point of P in its interior. Hence, we can construct a disk with pq as the diameter. This disk lies entirely in $lune(p, q)$ and is consequently empty as well. Thus, every such edge pq satisfies the **edge characterization** property. Therefore, it lies in the Delaunay Triangulation of P. Hence, if an edge belongs to RNG of a point set, it will belong to its Delaunay Triangulation as well. Hence, we can conclude that the Delaunay graph contains the RNG of point set P.

7.3 (c)

- Compute the Delaunay Triangulation $\mathcal{T}_D$ for $\mathcal{P}$. The graph obtained contains $O(n)$ edges and vertices. This step takes $O(n \log n)$ time.
- For every edge PQ in $\mathcal{T}_D$, we need to check if $lune(P, Q)$ is empty or not. If it is, then PQ belongs to $RNG(\mathcal{P})$, otherwise it does not.
- We scan over every point one-by-one and check if it lies inside circle centred at P, $\mathcal{C}_P$. If it does, we check if it also lies in $\mathcal{C}_Q$. If the point in question lies in both the circles, then it lies in $lune(P, Q)$. In that case, we skip this edge, otherwise add it to the $RNG(\mathcal{P})$.
- It takes $O(1)$ time to check if a vertex lies in both $\mathcal{C}_P$ and $\mathcal{C}_Q$. We do this for every vertex for a particular edge. Hence, checking membership of an edge in $RNG(\mathcal{P})$ takes $O(n)$ time.
- Since, there are a total of $O(n)$ edges in $\mathcal{T}_D$, the overall time complexity of the algorithm is $O(n^2)$.

COMPUTE_RNG($\mathcal{P}$)

$\mathcal{T}_D$ = DELAUNAY_TRIANGULATION($\mathcal{P}$)
for every edge $PQ \in E(\mathcal{T}_D)$
    $\mathcal{C}_P$ is the circle centered at P with radius $d(P, Q)$
    $\mathcal{C}_Q$ is the circle centered at Q with radius $d(P, Q)$
    flag = $TRUE$
    for every vertex $v \in V(\mathcal{T}_D)$
        if v lies in $\mathcal{C}_P$ and $\mathcal{C}_Q$
            $lune(P, Q)$ is not-empty, so edge $e \notin inRNG(\mathcal{P})$
            flag = $FALSE$
            break
    if (flag == $TRUE$)
        Add PQ to $RNG(\mathcal{P})$