Computational Geometry: Homework 3 Solutions

1. Edge-maximal planar graphs.

   - Let e be an edge in an edge-maximal graph embedded in the plane such that every triangle containing e is a face of the embedding. Prove that an edge-maximal graph is still edge maximal after contracting an edge.

     Answer: Let V, E be the number of vertices and edges respectively of an edge-maximal planar graph G. When contracting an edge e, the decrease in the number of edges is one more than the number of triangles containing that edge. If e satisfies the condition above, contracting e forms a minor, G', with $V' = V - 1$ vertices and $E' = E-3$ edges. Planar graphs are edge maximal if and only if $E = 3V-6$. The new graph G' has $E' = E - 3 = 3V - 9 = 3V' - 6$, and so it is also edge-maximal.

   - Prove that plane triangulations with at least 4 vertices are 3 connected.

     Answer: Fix an embedding of the triangulation. First, we prove that every plane triangulation with $n \geq 4$ vertices has at least n edges e with the property that the only triangles containing e are faces. The proof is by induction on n. The base is case is K_4 and holds trivially because all 6 edges have this property. For the induction, pick any triangle t that is not a face. We split the graph into two parts, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where G_1 is the induced graph on t and all of the vertices inside t in the embedding and G_2 is the induced graph on t and all of the vertices outside t. Both graphs are plane triangulations and both have fewer than n vertices. So, by induction, there are $|V_1| + |V_2| = n + 3$ edges in the two graphs with the desired property. When we combine the graphs, the only edges that can lose this property are the edges of t because no edges pass between any of the other vertices of G_1 to any of the other vertices of G_2. Thus, at most 3 edges with the desired property are lost in the combining, leaving us with $|V_1| + |V_2| - 3 = n$ such edges, as desired.

     Now, we can prove that plane triangulations with at least 4 vertices are 3-connected. Suppose for contradiction that G is not 3-connected. Then there is a pair of vertices u, v that disconnect it.

Let X and Y be two disconnected induced subgraphs of $G \setminus \{u, v\}$. By the preceding claim, as long as there are at least 4 vertices, we can contract an edge. Repeatedly contract edges, keeping the graph edge-maximal, until only 4 vertices remain. We must be careful never to contract an edge between u and v and to keep around at least one vertex of X and at least one vertex of Y. The resulting graph is edge-maximal with 4 vertices and therefore must be K_4. However, the vertices u, v still disconnect the graph. Since K_4 is 3 connected, this is a contradiction.

2. Here is a slightly simplified version of a real problem that arises in wireless communication between vehicles.

   Suppose we have n cars in the plane. Each car will be represented by a unit disk. The antenna is located in the center of the disk. The disks are all disjoint. Two cars have an unobstructed line of sight if the line between their antennae does not intersect any other cars. We want to know which are visible from which. Give an $O(n \log n)$ algorithm for finding all lines of sight from a given car. This will yield an $O(n^2 \log n)$ time algorithm for all of the cars.

   Present your algorithm in pseudocode and prove that it is correct. Also give a proper runtime analysis.

   Answer: It suffices to find all the lines of sight from a particular car in $O(n \log n)$ time.

   There were two common solutions presented fro this problem. The two variants are dual to each other. One approach was to sweep a ray from one car like sonar and keep track of the intersections of the other cars with this ray. The other approach was to sweep a growing circle out from one car and keep track of the intersections of the other cars with this circle. Both solutions are sweeplines.

   viewing the plane with polar coordinates, the duality between the approaches comes from the choice of using r or θ as the time variable. In one case, events are ordered by angle and handled by distance (radius). In the other case, the events are ordered by distance and handled by angle.

   Here is a more detailed description of one of these solutions.

   We will use an interval tree that stores a collection of intervals. In $\log n$ time, it can tell if a query point α is contained in any interval. If any interval (α, β) inserted into T spans 2π, we will instead split it into two intervals, $(\alpha, 2\pi)$ and $(0, \beta - 2\pi)$.

   (a) Pick a center car c.

   (b) Sort the cars by distance to c.

   (c) Initialize an interval tree T.

   (d) Initialize an output stack S

(e) **For each** car d

- **Let** α be the angle of the segment $\overline{cd}$
- **if** T does not contain α then $S.push(d)$.
- **Let** I be the interval of $[0, 2\pi)$ subtended by d with c's antenna.
- $T.insert(I)$

Correctness: A car d is visible from c if and only if the line segment $\overline{cd}$ is unobstructed. This is the case iff the angle α of this segment is not be blocked by any other car closer to c than d. If there was such a car, then it was processed early because we sorted the cars by distance. Moreover, the set of angles it blocks were inserted into T. Thus T contains α if and only if d the view of d is obstructed. So, the algorithm outputs a car d if and only if it is visible from c.

Running Time: Sorting the cars takes $O(n \log n)$ time. We loop n times and perform 1 search and 1 insertion into the interval tree. These operations take $O(\log n)$ time each and so the total running time is $O(n \log n)$.

3. **The non-planarity of K_5 and $K_{3,3}$.** In class, we showed that if a planar graph is 3-connected, then the faces are just the non-separating cycles. Use this Theorem and Euler's formula to show that K_5 and $K_{3,3}$ are not planar.

   Answer: The graphs K_5 and $K_{3,3}$ are both 3-connected. So, if they were planar, their faces would be the non-separating cycles.

   Suppose for contradiction that K_5 is planar. Every triangle in K_5 is a non-separating cycle, so $F \geq 10$. However, Euler's formula implies that there should only be $10 - 5 + 2 = 7$ faces, a contradiction.

   Suppose for contradiction that $K_{3,3}$ is planar. Choosing two vertices from each part in the bipartition, induces a 4-cycle and the remaining pair of vertices are connected. So, there are at least 9 non-separating cycles and therefore at least 9 faces in any planar embedding of $K_{3,3}$. However, Euler's formula implies that there should only be $9 - 6 + 2 = 5$ faces, a contradiction.

4. Recall that the history DAG data structure for the incremental Delaunay triangulation algorithm works by keeping around the different triangulations created throughout the process. There is an edge between triangles that overlap and are contained in triangulations that differ by only one point. If a triangle has only one outgoing edge, we contract it.

   We gave an analysis of this structure in the case where the input order is random. In this problem, you'll give another analysis under an assumption that the Delaunay triangulation the divides the points nicely. This assumption is not exactly true, but it's close.

   Imagine it were the case that the expected number of points in a Delaunay ball (i.e. the circumball of a Delaunay triangle) of a subset of r

points is $O(n/r)$. Prove that the history DAG can do point location in expected $O(n \log n)$ time under this assumption.

Answer: As we saw in class, after the rth point is inserted, the expected number of new triangles (and thus the number of new HDAG nodes) is O(1). The number of times each if these nodes will be accessed in future point locations is $O(n/r)$. Thus, the total number of node accesses when inserting all n points is

$$\sum_{r=1}^{n} O(n/r) = O(n \log n).$$

Be careful here. This proof also uses assumes that the $O(n/r)$ points conflicting with a given Delaunay triangle is independent of the number of new triangles created by adding point r. Otherwise, we would not be able to argue that the expectation of the product is the product of the expectations.

For a more complicated proof that does not use the independence assumption, you could find one in the de Berg et al. book in Chapter 9.