COL 752 Geometric Algorithms
Major Sem I 2016 -17 Max 80, Time 2 1/2 hrs

Note (i) Attempt any five of the following questions. If you try more than five, I will only grade the first five attempted and not the best five. Not all questions carry equal marks.
(ii) Every algorithm must be first described as a short high level description of the basic idea followed by a structured format. There must be a proof of correctness and a formal analysis of running time and space bound.
(iii) Feel free to quote any result from the lectures without proof - for any anything new, you must prove it first.

1. Suppose we know that the number of intersections denoted by m induced within a given set of n segments is $\Omega(n^{3/2})$. Design a simple algorithm using random sampling that reports all the m intersections in $O(m)$ steps.
Note that the trivial algorithm checks for all pairwise intersections but we cannot afford to do it for all the n segments. Moreover the linesweep algorithm reports m itersections in time $O(m \log n)$. **(15)**

   Choose a sample R of size $O(\sqrt{n})$ and compute the trapezoidal map (vertical visibility map) of R, $\mathcal{T}(R)$. For each trapezoid $t \in \mathcal{T}(R)$, we find all the segments $S(t)$ intersecting t. For $S(t)$ segments we find the pairwise intersections by brute force. The running time for finding all the pairwise intersections is given by

$$\sum_{t \in \mathcal{T}} S^2(t)$$

   and the expected running time is given by $E[\sum_{t \in \mathcal{T}} S^2(t)]$ which is given by

$$\frac{n^2}{r^2} O(E[|\Pi^0(R)|])$$

   The number of trapezoids is given by $O(r + m\frac{r^2}{n^2})$ since the probability of choosing an intersection is r/n. For $r = O(\sqrt{n})$ this quantity is $O(\sqrt{n} + m/n)$ which is $O(m/n)$ for $m \geq n^{3/2}$. So the expected time for finding all the intersections is $\frac{n^2 \cdot m}{n \cdot n} = O(m)$.

   To construct the trapezoidal map and allocating the line segments, using brute force methods we take $O(m^2/n^2 + n \cdot \frac{m}{n})$ which is $O(m + m)$ since $m^2/n^2 \leq m \cdot \frac{m}{n^2} \leq m$.

2. Let $\mathcal{D}$ be a set of n disks of radius r. Design a linear time algorithm to find out if there are any pair of intersecting disks (only the decision version), assuming floor and ceiling operations can be performed in $O(1)$ times. Describe the associated data structure in details. **(10)**

   If two disks intersect, it implies that their centers are closer than $2r$. So the problem is to determine if the closest pair has distance less than $2r$. In a $2r \times 2r$ grid, each square can contain at most 4 points whose distances are less than $2r$. If the closest pair has distance less than $2r$ then they must be in the same square or in nighbouring squares where a square has 8 neighbours. We map the centers to this grid (actually hash them using ceiling function) and check for this condition by inspecting $O(1)$ neighbouring grids. As there are at most $O(n)$ squares to inspect, the algorithm takes linear time.

3. Given a set of n points on the plane, design an algorithm to connect the points into a simple polygon (no self-intersecting edges). Prove a matching lower bound for this problem. **(10)**

   Sort the points along x direction, and let p_1 and p_2 be the leftmost and rightmost points. Let P_u and P_l be the points above and below the line $\overline{p_1, p_2}$. Join the points in P_u (P_l) which are monotone chains without any intersections and define a simple polygon.

For the lower bound, we reduce the sorting problem to this problem. Given points x_i $1 \leq i \leq n$, map them to points (x_i, x_i^2) and construct a simple polygon, say $\mathcal{P}$. The ordered points of $\mathcal{P}$ are sorted along x axis. Note that there is the unique simple polygon for the set of points - any other ordering would lead to self intersections.

4. Given a set of n line segments (possibly intersecting), design a data structure that supports queries of the following kind
   (i) For any given line ℓ how many segments are intersected by ℓ ?
   The data structure should have size polynomial in n and the query time should be $O(\log n)$.

   Consider the $2n$ end points of the n line segments and denote this set by P. It is easy to argue that two lines ℓ_1 and ℓ_2 intersect distinct set of line segments iff they partition the points in P differently. (The converse is not true.) Since there are $O(n^2)$ partitions, compute the set of segments that a line intersects corresponding to each partition. From our previous observation, two different faces may correspond to same set of segments stabbed by a line. This can be stored along with the partition or equivalently with each face of the dual arrangement. Build a point location data structure in the dual plane that supports $O(\log n)$ query time and outputs the number of segments intersected.

   (ii) Find out if there is a line ℓ that intersects all the given segments - this line is called a *transversal*. **(10 + 5)**

   If there is a face that corresponds to count of n then there exists a transversal. Clearly this can be done in polynomial time.

5. Let R and B denote two sets having m and n points respectively. If R and B are linearly separable, show that R and B can be simultaneously bisected by a single line. You may assume that n and m are even integers. **(15)**
   Hint: You may assume the line separating R and B to be the y-axis wlog and use duality. Consider the duals of the set B and R respectively - denote them by $B*$ and $R*$ respectively. The points in between the $n/2$ and $n/2+1$ levels in $B*$ correspond to the *halving partitions* of B - denote this by L_B. For the analogous points in R, denote this by L_R. The crux of the problem is to show that L_B and R_B intersect, whose dual (a line) simultaneously halves B and R.

   For this, note that the lines corresponding to B and R have negative and positive slopes respectively. Now you must argue that two levels, one of which consists of lines with negative slopes and the other one consists of positively sloping lines must intersect. Note that the levels are x-monotone and they separate the plane into disconnected regions. Consider, the middle level L_B^m coresponding to set B which is decreasing from $+\infty$ to $-\infty$. At $x - \infty$, the middle level, L_R^m of R is below L_B^m and at $x = +\infty$ it is above L_B^m, so they must interect somewhere.

6. Given a set of blue and red points (total number of points in n), find out if they are linearly separable in $O(n)$ time. **(10)**

   In the dual plane, it corresponds to a linear program where the blue and red points define intersections of the corresponding half-planes.

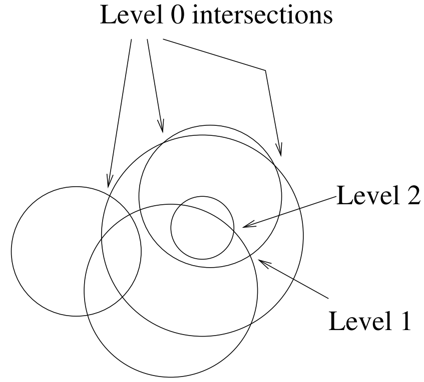
   Linear program in two dimensions can be solved in linear time.

7. Given a set S of n points in a plane and any $0 < \varepsilon < 1$, describe a linear time algorithm for finding a pair $p, q \in S$, such that
   $$d(p,q) \geq (1-\varepsilon) \max_{s,t \in S} d(s,t) \text{ where } d() \text{ is the Euclidean distance}$$
   Note that p, q gives an approximate diameter of S. **(15)**
   Consider the output of an ϵ WSPD algorithm - there are a linear number of pairs of subsets (A_i, B_i). Choose arbitrary representative from each subset ($rep(A_i)$ and $rep(B_i)$ and output the furthest rep pairs, say (p, q).

Suppose (s,t) be a diametral pair and suppose they belong to the subset pair (A_f, B_f). Then from triangle inequality $d(s,t) \leq d(rep(A_f), rep(B_f)) + d(s, rep(A_f)) + d(t, rep(B_f)) \leq (1+2\epsilon) \cdot cd(rep(A_f), rep(B_f)) \leq (1+2\epsilon) d(p,q)$.
Or in other words $d(p,q) \geq \frac{1}{1+2\epsilon} \cdot d(s,t)$. By choosing $\epsilon = \varepsilon/2$, the resut follows.

8. Given a set of n disks (possibly of different radii), they partition the planes into maximal connected regions. A point p has depth d ($0 \leq d \leq n$) if it is contained within d disks.

   Consider all the pairwise intersection points of the disks - there are at most $O(n^2)$ such points. We will consider the depth of the intersection points between 0 to $n-1$.
   Derive a bound on the maximum number of points having depth k. **(20)**
   Hint: Use lifting transform to bound level 0.



   Using the lifting transfor $z = x^2 + y^2$, it is known that the a point $p = (x', y')$ is inside/outside a disk of radius r iff the lifted point is above/below the plane passing through lifted disk. (This was worked out in the lecture on Delaunay disks). Therefore the level 0 vertices must be on level 0 of arrangements of planes in three dimensions. This level has size $O(n)$ which corresponds to the size of intersection of n half-planes. The level zero intersectin points corespond to the intersection of the paraboloid with the lines of this arrangement and therefore they can be bounded by $O(n)$.

   It can be shown using calculations identical to the bound on $\leq k$ levels of two dimensional arrangements that the number of vertices in levels $\leq k$ is boundedby $O(k \cdot n)$ and hence the k-th level is also bounded by $O(kn)$.