CSL 852 Computational Geometry

Major , Sem I 2010-11, Max 80, Time 2 hrs 30 minutes

Note (i) Write your answers neatly and precisely. You won't get a second chance to explain what you have written. (ii) Every algorithm must be accompanied by proof of correctness and a formal analysis of running time and space bound. Feel free to quote any result from the lectures without proof - for any anything new, you must prove it first. (iii) All questions carry equal marks.

1. Let $\mathcal{D}$ be a set of n disks of radius r.

   (a) 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.

   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.

   (b) If there there is no overlap, describe an algorithm to to report the closest pair of disks in $O(n \log n)$ time using randomized incremental construction.

   We add the points in a random order and maintain closest pair. The probability that a closest pair has to be updated is less than $2/i$ for the i-th step by backward analysis as the closest pair is defined by two points. We use the previous data structure (a grid of size of the closest pair) to check this and rebuild it if the closest pair has changed. The expected time for each step is $2/i \cdot i = O(1)$ if we use hashing or $O(\log n)$ if we store the squares in a dictionary. Thus the total expected running time is $O(n \log n)$.

2. Given a set S of n points in the plane, design a data structure that returns *number of* points that lie within distance D (for some fixed D) from a given point q. The query should be answered in $O(\text{polylog}(n))$ time and the size of the data structure should be polynomial in n.

   Construct the arrangement of n circles of radius D centred at the given points. For each cell of the arrangement, the answer to the query q is fixed (exactly those points whose circles contain q. You must show how to preprocess the circle arrangement to do point location in polylog time.

   Otherwise, you argue on the basis of the lifting transform, tha the inside and outside of the disks are mapped to below and above a hyperplane. So the number of intersecting disks at any given point is the number of planes (in 3d) above the vertical projection.

   For this you must show how to preprocess a set of planes in 3d to do point location in polylog time. *This was not done explicitly in the lectures.*

3. A partition of S into two subsets S_1 and S_2 is called *linearly separable* if there is a line such that S_1 lies on one side of the line and S_2 on the other side.

(a) Give a tight bound on the number of linearly separable partitions of a set of n points in the plane.

In the dual space, look the arrangements of lines (corresponding to the input points). All the points in a face correspond to a partitioning line in the original space corresponding to the same partition. Since there are $O(n^2)$ faces there are $O(n^2)$ distinct partitions. This bound is achievable as one can argue using a convex $n-gon$.

(b) 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.
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 can argue that two levels, one of which consists of lines with negative slopes and the other one consists of positively sloping lines must intersect.

4. Let $S = \{p_1, p_2 \dots p_n\}$ be a set of n points in the plane and let k be a positive integer.We would like to cover all the points using k disks of diameter D, such that D is minimum - let it be denoted by D_o.

(a) Let $q_1, q_2 \dots q_{k+1}$ be a subset of S such that for all $j \geq 1$, $d_j \geq d_{j+1}$ where $d_j = \max_i ||q_{j+1} - q_i|| \quad i = \{1, 2 \dots j\}$.
Argue that the $D_o \geq ||q_{k+1} - q_k||$.

Consider the alternate definition that the points in $q_1 \dots q_{k+1}$ are separated at least by distance d. Then any optimal solution will consist of k disks such that some disk must contain at least 2 points (from pigeon hole argument). Therefore $D_o \geq d$.

(b) Let the points q_i be defined in the following way. Start from an arbitrary point $q_1 \in S$. Let q_2 be the furthest point from q_1 and for $1 < i \leq k+1$, $q_i = \max_{p \in S} d(p, Q_{i-1})$ where $Q_{i-1} = \{q_1, q_2 \dots q_{i-1}\}$ and $d(p, Q)$ denotes the distance from p to its closest neighbour in Q.
Prove that you can cover the points of S using k disks of radius D_o, thereby obtaining a factor 2 approximation algorithm.

Clearly all points are within a distance of d_k from Q_{k-1} (as d_k is the furthest distance neighbour of Q_{k-1}). So the disks of radius d_k (diameter $2d_k$) centered at Q_{k-1} cover all points.

To prove the approximation bound, first you must **prove** that the points q_i defined this way satisfy the conditions of the part (a). Let $d_{i-1} = d(q_i, Q_{i-1}) \quad i > 1$. By induction show that $d_{j+1} \leq d_j$. So, $D_o \geq d_k$ and the covering disks have diameter $2d_k$.

5. The weight of an εWSPD is defined as $\sum_i^k(|A_i| + |B_i|)$ where $\{(A_1, B_1) \dots (A_k, B_k)\}$ are the pairs of the WSPD. For a fixed ε, construct a set of points P such that the weight of a valid WSPD of P has weight $\Omega(n^2)$.

Given ε, consider n points on a line at coordinates $\alpha, 2\alpha \dots 2^{n-1}\alpha$ where $\alpha = \lceil 1/\varepsilon \rceil$. We have to show that for this configuration of points, the WSPD will always have weight $\Omega(n^2)$ whatever be the algorithm used. Let p_1, p_2 be the points in increasing order.

For this point set, all the WSP (A_i, B_i) must be such that they are contained in non-overlapping intervals (otherwise distance is 0). Moreover, if the points in B are larger than A then $|B| = 1$. Otherwise the diameter is larger than the separation. The weight of such a pair is $|A_i|$ and it covers exactly $|A_i|$ pairs, therefore the total weight must be at least $\Omega(n^2)$ for covering all pairs.

Note that it is not sufficient to show that some WSPD has weight $\Omega(n^2)$ as that is always true for the trivial WSPD.