COV 866 Special Module in Algorithms
Final Exam Sem II 2017 -18 Max 75, Time 2 hrs

(i) You can quote any result presented in the lectures without proof. For anything else, you must provide the proof.
(ii) For problems 3 and 4, you can attempt them as a take-home assignment problem for half the credit (to be submitted by 19th April noon).

1. Consider a set S of 3 dimensional points. We want to find the smallest enclosing ball $\mathcal{B}$ containing the points. Design an efficient algorithm for n points and fixed dimension using RIC. **(15)**

   The smallest enclosing ball is defined by at most $d+1$ points on its boundary or by a pair of diametral points. In our case $d = 3$. We add the points in a random order and maintain the smallest ball B_i^d for the first i points in d dimensions. When the $i+1$st point is added two cases arise -
   $p_{i+1} \in B_i^d$: Then $B_{i+1}^d = B_i^d$ else
   $p_{i+1} \notin B_i^d$ We have to construct the ball B_{i+1}^d. In this case we know that $p_{i+1} \in B_{i+1}^d$, so we need to find the remaining d points from i points.

   Let $T(i, j)$ denote the time to find the smallest ball containing i points given that $j \leq d+1$ points have to be found that define the smallest ball. The overall running time can be denoted by $T(n, d+1)$. From backward analysis, we know that the expected time for the i-th step is $\frac{j}{i} \cdot T(i-1, j-1)$. If $T(i, j)$ is the expected running time for the RIC, then the recurrence is similar to that of linear programming and the expected running time is given by $O((d+1)!n)$.

2. If Y is an ε sample of range space $(X, \mathcal{R})$ and P is a δ net of $(Y, \mathcal{R}_{|Y})$, then P is an $\varepsilon + \delta$-net of X.
   Note that this can be used to construct an ε net by choosing $\delta = \varepsilon$ using straightforward random sampling since the range space (corresponding to Y) is much smaller. This will not have $\log |X|$ factor. **(10)**

   Let d be the VC dimension of $(X, \mathcal{R})$, where $|X| = n$. A δ-net of Y has the property that it will pick at least one point from a range of Y that has more than $\delta \cdot |Y|$ points.

   Consider any range $r \in \mathcal{R}$ that has more than $(\varepsilon + \delta) \cdot n$ points. Then r must satisfy

$$|r \cap Y| \geq |Y| \cdot \frac{|r \cap X|}{|X|} - \varepsilon \cdot |Y| = |Y|((\varepsilon + \delta) - \varepsilon)) = \delta |Y|.$$

   So $|P \cap r| \geq 1$ and hence it is a $\varepsilon + \delta$-net of X.
   Note that Y has size $y = O(\frac{d \log(1/\varepsilon)}{\varepsilon^2})$ and $\mathcal{R}_{|Y}$ has size y^d.

3. Given a set of n lines $\mathcal{L} = \{\ell_1, \ell_2 \ldots \ell_n\}$ (assume that that no three lines are concurrent), an $\frac{1}{r}$ *cutting* of $\mathcal{L}$ is a triangulation Ξ of the the plane such that
   (i) No triangle in Ξ intersects more than $O(\frac{n}{r})$ lines of $\mathcal{L}$. Note that $r \geq 1$.
   (ii) Ξ contains $O(r^2)$ triangles.

   (a) Can Ξ contain fewer than r^2 triangles (asymptotically) ?

   Since n lines induce $\binom{n}{2}$ intersections, suppose there are t triangles. Then some triangle will contain at least $\Omega(\frac{n^2}{t})$ intersections, implying that there are $\Omega(\frac{n}{\sqrt{t}})$ lines in the triangle. Since, it is required that $\frac{n}{\sqrt{t}} \leq \frac{n}{r}$, it implies that $\sqrt{t} \geq r$ or equivalently $t \geq r^2$.

   (b) Describe an algorithms to construct Ξ efficiently ?
   Hint: A random sample of r lines can be converted to a trapezoidal map - use random sampling lemmas on the trapezoids. If the bound exceeds $O(n/r)$ in any trapezoid τ_i choose an additional sample r_i in τ_i to achieve the bound of $O(n/r)$. Now bound the total additional trapezoids created by the r_is over all the trapezoids so that we don't exceed $O(r^2)$ triangles overall . **(5+15)**

Choose a sample of size $r \leq n$ and construct the trapezoidal map where the intersection points also have vertical lines passing through them that define trapezoids (notice that there are no end-points). So there will be $O(r^2)$ trapezoids and we can invoke the result on bounding the number of unsampled lines intersecting every trapezoid by $O(\frac{n \log r}{r})$ with probability exceeding $1 - 1/r$ (if not repeat sampling). Now consider each triangle Δ_i that contains $t_i \cdot \frac{n}{r}$ lines for $t_i > c$ for some integer c. (For triangles that intersect $\leq c\frac{n}{r}$ lines, they already satisfy the requirement.) For such trinagles, choose a sample of size $t_i \log(t_i)$ and using this sample further subdivide Δ_i into trapezoids - about $O(t_i^2 \log^2 t_i) \leq t_i^3$ trapezoids. Using the bound on maximum number of intersections, it follows that no trapezoid has more than $t_i \cdot \frac{n}{r} \log(t_i \log t_i)/(t_i \log(t_i)$ lines which can be bound by $O(n/r)$ lines. This satisfies the requirements but we have introduced more triangles that can be bounded as follows. Let n_i denote the number of lines intersecting Δ_i.

$$\sum_i t_i^3 \leq \sum_i \frac{n_i^3}{n^3/r^3} = \frac{r^3}{n^3} \sum_i n_i^3 = \frac{r^3}{n^3} \cdot \frac{n^3}{r^3} \cdot O(r^2) = O(r^2).$$

The second last equality follows from the generalized bound of the sum of the cubes of the subproblems. i.e. $E[T_3]$. Although it is an expected bound, the sampling can be repeated till we obtain the right bounds.

4. Given a set P of n points on the plane and a set L of $m = n^2$ lines, we want to construct a spanning tree of P such that no line $\ell \in L$ intersects more than $O(\sqrt{n})$ tree edges. Note that a spanning tree can be constructed by using any of the classical algorithms on a graph whose edges are defined as the segment connecting p_1, p_2. For this, we use the following procedure. We compute one edge of the spanning tree in each round where the number of connected components in a round is bounded by $n - i$ for $0 \leq i \leq n-1$. We choose one representative point say C_j $j \leq n - i$ in round i. Initially we have n components (isolated points).

   We construct a $\frac{\alpha}{r_i}$ cutting with respect to L where $r_i = \sqrt{n-i-1}$ and a suitable constant $\alpha \geq 1$. This gives us a set of at most r_i^2 triangles (α was chosen so that we have no more triangles in the cutting) where each triangle intersects no more than $\frac{\alpha m}{\cdot r_i}$ lines from L. So some triangle must have atleast 2 points C_k and C_l. Include the edge connecting these two points. Clearly it doesn't intersect more than $\frac{\alpha \cdot m}{r_i}$ lines of L.

   Double the weights of the lines intersecting the chosen edge and repeat the process. Note that we use weighted version of the cutting theorem where the initial weights are 1 and subsequently modified as above.
   (1) Show that no line $\ell \in L$ intersect more than $O(\sqrt{n})$ lines of the MST produced by the above procedure. **(15)**

   The weighted version the $\frac{1}{r}$ cutting theorem is such that given an weight function $f : L \Rightarrow \mathbb{R}$ such that $\sum_{\ell \in L} f(\ell) = W$ then we can produce a set of triangles $\mathcal{T}$ such that the weight of the lines intersecting any triangle $t \in \mathcal{T}$ is bounded by W/r.

   So, the weight of the lines in each iteration can increase by a factor $(1 + \frac{1}{r})$. After i iterations the total weight of the lines $W_i \leq (1 + 1/r)^i \cdot W$. The weight of any line after k doublings is 2^k and it intersects at most k edges of the spanning tree. So

$$\prod_{i=1}^{n-1} (1 + 1/r_i) \cdot W \geq W_i \geq 2^k$$

   Equivalently $\prod_i e^{1/r_i} \geq 2^k$ or $e^{\sum_i 1/r_i} \geq 2^k$. Since $\sum_i 1/r_i \leq O(\sqrt{n})$, $e^{O(\sqrt{n})} \geq 2^k$ or $k = O(\sqrt{n})$.

   (2) Can we generalize this to any line ℓ as opposed to $\ell \in L$? **(5)**

   The number of *combinatorially distinct* lines that intersects distinct set of edges of the spanning tree is $O(n^2)$. This follows from the distinct ways that the n points can be partitioned by a straight line (think in terms of the dual transformation). So by choosing one representative of each of the possible partitioning lines in L, we can ensure this property for any line on the plane.

(3) Show that there exists a set P such that there will be some line that intersects at least $\sqrt{n}$ edges. **(10)**

Consider a $\sqrt{n} \times \sqrt{n}$ grid. Put n points, one per square. If you consider the $\sqrt{n}$ horizontal and $\sqrt{n}$ vertical lines, there must be at least $n-1$ intersections with any spanning tree (every point is connected to some other point). So some line must have at least $\Omega(\sqrt{n})$ intersections.

5. Submit Problems 2,3, 5 from assignment sheet 2 by April 23rd..