Department of Computer Science and Engineering: IIT Kharagpur

Computational Geometry: CS60064: End-semester examination : Spring 2013
Instructor: S P Pal: Maximum marks: 100: There is limited choice.

[1] Define the *beach line* and explain how the *sweepline algorithm* for computing the Voronoi diagram of n point sites in the plane computes the vertices of the Voronoi diagram. [7+8 marks]

[2] Design data structures for the (i) preprocessing, and (ii) query processing steps for the following problem. Given an n-sided convex polygon P, create a data stucture of $O(n)$ size in the preprocessing step in $O(n)$ time, so that for any *query* straight line L in the plane, it is possible to check in $O(\log n)$ time whether L intersects P. [8+7 marks]

[3] State the *incremental randomized algorithm* for 2-dimensional *Linear Programming* where n constraints are provided. Show that a naive deterministic solution for processing the constraints in incremental fashion may lend a $O(n^2)$ time method for computing the optimal solution. Derive an upper bound on the probability that there will be a change in the *optimal solution point* in the kth iteration, where $3 \leq k \leq n$, when k constraints have been processed. Determine the expected running time of the entire algorithm. [3+3+5+4 marks]

[4] Characterize (i) the vertices of the Voronoi diagram, and (ii) the edges of the Voronoi diagram using *empty circle* properties. In particular, suppose a set S of n points in the plane forms a convex polygon of n sides. Characterize the Voronoi diagram for such a set S of points? [7+8 marks]

[5] Let S be a set of n line segments in the plane with a total of k intersecting pairs of segments. Clearly, $k = O(n^2)$. Let us consider a *random permutation* of segments $< s_1, s_2, ..., s_n >$ in S. Assume that each such

permutation is equally likely when we choose such a random permutation. Let S_j be the set of the first j segments of the random permutation, and k_j be the number of intersecting pairs of segments in S_j. Show that expectation $E(k_j) \doteq O(k\frac{j^2}{n^2})$, where $E(k_j)$ is computed over all possible random permutations of segments in S.

Show that the problem of computing the planar map of the n line segments in S has time complexity $\Omega(n \log n)$. [10+5 marks]

[6] Given a set L of n lines in the plane, consider the arrangement $A(R)$ of r lines from a random sample $R \subset L$. Here, all the possible combinations of n lines taken r at a time are equally probable for the set R. Partition each face of $A(R)$ by drawing vertical segments from intersection points, giving rise to a planar subdivision $A'(R)$, which has only trapezoids and triangles. Given a set of m faces of $A(R)$, show how we can use (i) *Canham's bound* in each cell of $A'(R)$ for the $n_i \leq n - r$ lines of S intersecting the ith cell of $A'(R)$, and (ii) the *zone theorem* for the r lines, in order to estimate the *many faces complexity* for the m faces in $A(R)$. How is the *probabilistic method* used for estimating an upper bound on the worst case *many faces complexity* $K(m, n)$, for m cells in an arrangement of n lines? Assume that the *expectation* of $\sum m_i n_i^{\frac{1}{2}}$ over all cells of $A'(R)$ is $O(mn^{\frac{1}{2}})$. [5+5+5 marks]

[7] Show that *randomized quick sort* runs in expected $O(n \log n)$ time as follows. Use a random permutation of the unsorted input of n real numbers. Process one number at a time from the random permutation sequence by splitting an existing interval. When $i + 1$ points are already processed, and $i + 2$ intervals are created, show that the probability that any (specific) real number (of the remaining $n - i - 1$ real numbers) required an *update* during the $(i + 1)$st insertion is at most $\frac{2}{i+1}$. [15 marks]

[8] State in details how linear time triangulation can be done for (i) monotone polygons, and (ii) polygons where every interior point is visible from some point of a given boundary edge of the poygon. [8+7 marks]

[9] Show that given two points s and t inside an n-sided simple polygon P, the shortest Euclidean path $SP(s, t)$, can be computed in linear time, provided a triangulation of P is given. Show also that $O(n \log n)$ time suffices if we wish to find $SP(s, v_i)$, for all vertices v_i of P. [8+7 marks]

==========================