Department of Computer Science and Engineering
IIT Kharagpur
April 2012: Time : 3 hours: End-semester examination
Computational Geometry: CS60064 (PG) Instructor: S P Pal: Maximum marks: 100
There is limited choice.

1. 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 and insert one number at a time from the random permutation sequence. 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 $\frac{2}{i+1}$. Now show that the expected running time is $O(n \log n)$ using these probabilities. [10 marks]

2. Show that a $\frac{1}{r}$-cutting of $O(r^2 \log^2 n)$ triangles is possible for $n$ given lines in the plane. [10 marks]

3. Outline the use of the Kovari, Sos and Turan extremal graph theory result, and derive the sub-optimal *many faces* complexity bound of $O(m\sqrt{n} + n)$, for $n$ lines and $m$ faces using the extremal result.

   Use (i) the zone theorem at the borders of the cells defined by the trapezoidal decomposition of the arrangement defined by a random sample of $r$ out of the $n$ lines, and (ii) the above sub-optimal bound, in order to present an outline of the derivation of the optimal $O(m^{\frac{2}{3}} n^{\frac{2}{3}} + n)$ *many faces* complexity bound. [8+12 marks]

4. We need to preprocess a set $S$ of $n$ line segments in the plane, each of which may have an arbitrary slope, so that given a finite vertical query segment $s$ joining points $(x, y)$ and $(x, y + c)$, we can report all the segments in $S$ that intersect the query segment $s$. We require to use *segment trees* and *range trees.* Design your scheme for the best possible (polynomial) preprocessing, and (output sensitive) query complexities. [15 marks]

5. Given a *funnel* whose base is a boundary edge of a simple polygon, show how the region inside the funnel can be triangulated using *diagonals* of the polygon. Can we triangulate all the funnels in linear time? Why? [A diagonal is a line segment joining two vertices (of the polygon) that does not intersect the exterior of the polygon.] [8+7 marks]

6. How can we triangulate monotone polygons? Give details of linear time triangulation of such polygons. [15 marks]

7. Consider the randomized and incremental construction of *binary space partition* trees *(BSP trees)* for $n$ given lines segments in the plane, considering only *auto-partitions.* The expected number of fragments of these $n$ segments generated during the construction of the BSP tree is an important performance parameter. Show that the expected number of cuts generated by a segment is no more than $O(\log n)$. [10 marks]

8. Consider the *randomized incremental* two-dimensional linear programming algorithm where the constraints are processed in the order determined by a random permutation. Let $v_i$ denote the point that optimizes the objective function after $i$ constraints are processed. Show that the probability that $v_{i-1} \neq v_i$ is at most $\frac{2}{i}$. [10 marks]

9. Define *binary space partition* trees *(BSP trees)* for $n$ line segments in the plane where we consider only *auto-partitions.* How are such trees used to implement the *painter's algorithm* for rendering? [8+7 marks]

10. Given a set $S$ of points in the plane, sorted by x-coordinates, design a deterministic linear time algorithm for triangulating the point set $S$. The edges of the triangulation are intersection-free except for endpoints from the set $S$. [10 marks]

---