Indian Institute of Technology Kharagpur

Spring Semester 2021

COMPUTER SCIENCE AND ENGINEERING

Computational Geometry

Date: 03 April 2021 **Total points** = 100; Credit: 20%

Key to Online Test-03 **Time:** 11:00 AM - 12:30 PM

Instructions (Read carefully)**:** This is an MCQ or Fill_up_Gap type, OPEN-BOOK, OPEN-NOTES, test. For each **MCQ** (or Fill_up_Gap) type question, please choose one answer from the given choices (or write your answer). Each *correct answer* will fetch 5 points, *incorrect answer* will contribute 0 point, and *no answer* (full omit) leads to 1.5 point. Thus, if you have doubt, it may be beneficial to skip a question. If a question has multiple fill-up-gaps, its components will be graded proportionately. This question paper has four pages.

Submission of answers: Please create a text/pdf file including **your name, roll-number**, and your answers, and submit it to the CSE Moodle Page by 12:45 PM on 03 April 2021. For each question, **write your answer only**, details of solutions are **not** needed.

---

1. Consider an arrangement *A* of 100 lines in general positions. A new line *l* is now added to *A*. Assume that the arrangement is *simple*, i.e., no three lines are concurrent, no two lines are parallel, and no line is vertical).

The number of faces of *A* intercepted by *l* is (make the right choice below and/or fill-up the gap):

A. exactly 101;
B. may lie between ____ and _____ ;
C. statements such as A or B cannot be made, it will depend on relative slopes of 100 lines;
D. statements such as A or B cannot be made, it will depend on the position of *l*;
E. none of the above statements is applicable.

2. In the previous problem, let *F* denote the set of faces of *A* that have been intercepted by the new line *l*. The number of edges that surround all faces in *F* is at most (fill-up the gap): 600.

3. Given *n* lines in the 2D plane, the complexities of constructing an arrangement $A(n)$ are given by (choose the tightest one):

A. $O(n^2 \log n)$ time and $O(n^2)$ space;
B. $O(n^2)$ time and $O(n^2 \log n)$ space;
C. $O(n^2)$ time and $O(n^2)$ space;
D. $\Theta(n^2)$ time and $\Theta(n^2)$ space;
E. None of the above.

4. Consider a simple arrangement *A* of 50 lines in the primal and let *l* be one of these lines. Construct the dual of *A*, i.e., *A**, in the dual plane. (Fill-up the gaps): The number of unbounded faces in *A* is 100, and the degree of *l** in the planar sub-division of *A** will be 98. (By degree of a node in a planar sub-division, which is a planar graph, it is meant the # edges incident on it, not the #lines passing through it).

5. Let N be the set of all intersection points of a simple arrangement A of n lines and let $C(N)$ denote the convex hull of N. Let $N' = N \setminus C(N)$. The convex hull of N' can be computed in (choose the tightest one):

A. $O(n^2 \log n)$ time and $O(n^2)$ space;
B. $O(n^2)$ time and $O(n^2)$ space;
C. $\Theta(n^2)$ time and $\Theta(n^2)$ space;
D. $O(n \log n)$ time and $O(n)$ space;
E. None of the above.

6. Let p^* denote the dual of point p, and l^* denote the dual of line l. Consider two points p_1 and p_2 both in the first quadrant of the primal such that $x(p_1) < x(p_2)$, and $y(p_1) < y(p_2)$. Line l has positive slope and intersects with the segment (p_1, p_2) in the primal.
Fill-up the gap:
The relative position of l^* with respect to p_1^*, p_2^* in the dual plane, would be as follows: line p_1^* and line p_2^* will be intersecting with slope(p_2^*) > slope(p_1^*); point l^* would lie above p_1^* and below p_2^*.

7. Consider a simple arrangement $A(5)$ of five lines. We construct an undirected graph G such that
(i) each intersection point of $A(5)$ corresponds to a vertex of G; and
(ii) there will be an edge between two vertices v_1 and v_2 of G, if and only if the two intersection points corresponding to v_1 and v_2 do not share a line in $A(5)$.
The minimum length of a cycle in G is (choose one):

A. 3; B. 4; C. 5; D. 6; E. none of these

8. Let $x, y > 0$, and consider two points $p_1(x, y)$ and $p_2(-x, y)$ such that p_1 is incident on p_1^*, and p_2 is incident on p_2^*, when both the primal and dual plane are drawn on the same coordinate system. Let k denote the intersection point of p_1^* and p_2^*. Then the slope of k^* is (fill-up the gap): zero.

9. We have a top-open rectangular box randomly filled with n red balls and n blue balls as shown in Figure 1. The imbalance of a line l is defined as the difference in the number of red and blue balls that appear below l. The *offset* is defined as the supremum of imbalance over all positions of l within the box.

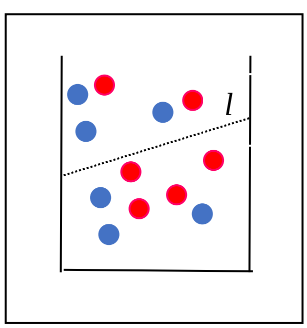

Figure 1: Distribution of bichromatic balls in a box

(a) (Fill-up the gap): The imbalance of a line l is equivalent to the difference in the #red and #blue lines above the point l^* , in the dual plane.

(b) (Fill-up the gap): The offset can be computed in $O(n^2)$ time and in $O(n^2)$ space (write the best known complexity).

10. Given a random distribution of n red balls and n blue balls in the 2D-plane, let l be a line that bisects both the sets into two equal halves. By Ham-Sandwich Theorem, such a line always exists. (Fill-up the gap): Finding l is equivalent to determination of the intersection point of the median levels for the red lines and that of blue lines, in the dual plane, and can be accomplished in $O(n)$ time (write the best known complexity).

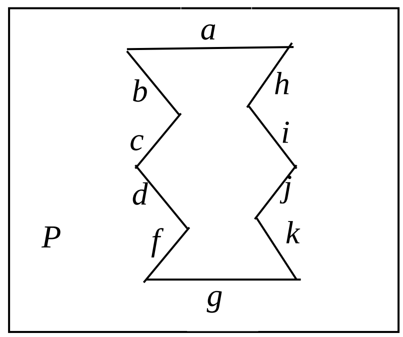

Figure 2: Mold for casting

11. Consider the 2D polygon P as shown Figure 2 above with ten facets $\{a, b, c, \ldots, k\}$. (Fill-up the gap): The top facet with respect to which P is castable in a single mold via translation and/or rotation is: φ (no top facet will work).

12. (Fill-up the gap): Given a 2D polygon P with n facets and a given top facet f, checking whether P is single-mold castable with respect to f via translation is equivalent to checking the existence of common intersection of n half-lines, and can be ascertained in $O(n)$ time (write the best known complexity).

Question 13, 14, 15 stated below, refer to the following problem description.

Eight points with the following coordinates are scattered on the 2D X-Y plane: {(17, 5) (21, 49), (12, 3), (7, 10), (15, 73), (2, 19) (5, 68), (8, 37)}. We want to perform an orthogonal range query within a box whose bottom-left corner is at coordinate (6, 8) and top-right corner is at (16, 40), using *range tree*. Answer the following three questions.

13. (Fill-up the gap): The total number of nodes (including leaves) in the X-tree is 15, and that of all Y-trees is 49. (Note: construction of Y-trees does not depend on the particular query-box).

14. Let α denote the internal node in the X-tree, whose label is 15. Assume that node β appears at the root of the Y-tree pointed by α. Then the label of β will be (choose one):

A. 73; B. 5; C. 49; D. 17; E. None of these

15. (Fill-up the gap): Given the above query-box, the number of canonical sets that are fully included in the search interval of the *X*-tree is 2, and the points that are returned following a search in the *Y*-tree are (7, 10) and (8, 37).

16. Given a set *S* of *n* points in a plane, a *kd*-tree has been constructed so that 2D range queries can be efficiently answered. Now, an axis-parallel box *B* is given, and you are asked to report the number of points in *S* that are spanned by *B*. The time complexity of such a query is $O(\sqrt{n})$, pre-processing time is $O(n \log n)$, and space is $O(n)$.

17. In the following 2D orthogonal range-query scenario implemented as a *kd*-tree, an axis-parallel query-box colored in *red* has appeared as shown in Figure 3 below.

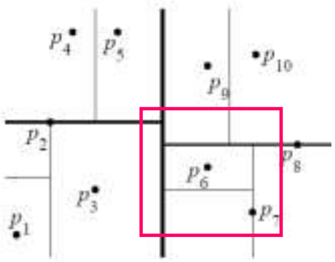

Figure 3: Orthogonal range query using *kd*-tree

(Fill-up the gap): The number of nodes in the *kd*-tree explored during the search procedure (including both internal and leaf nodes) is 15.

18. Five sorted arrays A_i (i = 1, 2 …, 5) of each having 2000 distinct data-values, are given. Furthermore, there is no common data between two arrays. Given a search key k, we have to return the smallest j such that $A_i(j) \geq k$, for all i. Fractional cascading has been deployed in the data structure to expedite the search.
(Fill-up the gap): The number of down-pointers and right-pointers that are to be set in the data structure will be 14,125 and 6125, respectively.

19. Consider an axis-parallel rectangular region *R*, where locations of *n* cities are given. The weight W_i associated with city *i* denotes the number of people infected with an epidemic there. Given an axis-parallel query rectangle *M* of fixed size, where the dimensions of *M* are much less than those of *R*, we want to determine the location of *M* inside *R*, which contains the maximum number of infected people. (Fill-up the gap):
The above problem is equivalent to determining the maximum-weighted clique in a rectangle-intersection graph, with weights (#people infected) attached to each vertex (corresponding to a city), and can be solved in $O(n \log n)$ time (give the best known complexity).

20. You are given a rectangle *R*(ABCD) whose vertices are ordered CCW. The length of *R* is very large, and width AD = BC = 10 unit. *R* contains *n* points such that the distance between every pair of points in *R* is at least 10. There is a query point *q* outside *R* facing the side AB. A point *p* in *R* such that distance(p, q) < 10, if present, can be determined in time (choose the best one):

A. $O(1)$ B. $O(\log n)$* C. $O(\log^2 n)$ D. $O(n)$ E. None of these

*Choice B. - multiple queries can be answered in $O(\log n)$ time each, with pre-processing time of $O(n \log n)$; $(\log n + k)$ will not be needed here since $k \leq 6$ due to rectangular packing).
Choice D. $O(n)$, may also be accepted as correct answer if one assumes that preprocessing is precluded.