CS60064 Spring 2022

Computational Geometry

Instructors

Bhargab B. Bhattacharya
Partha Bhowmick

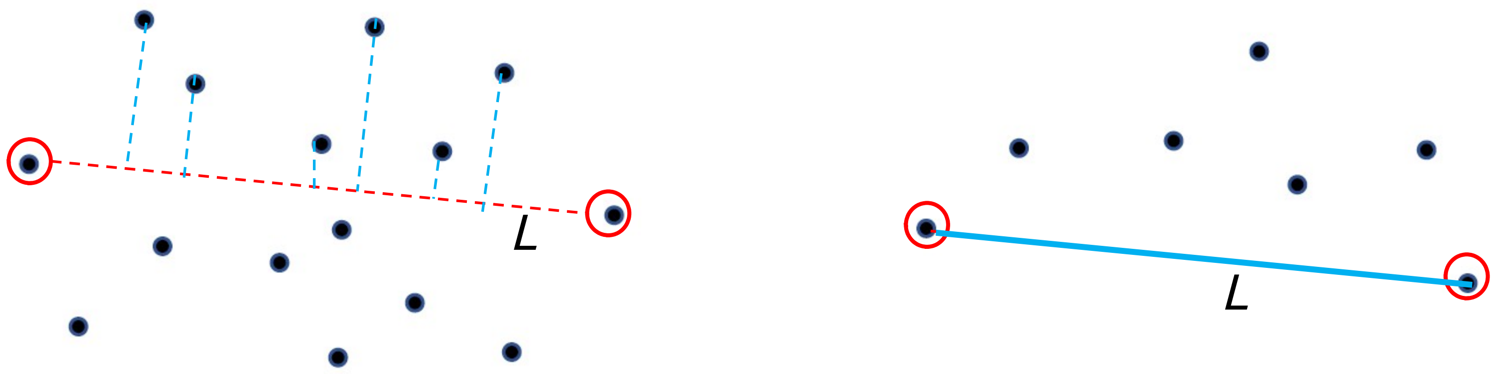
23 February 2022
Lecture #21

Indian Institute of Technology Kharagpur
Computer Science and Engineering

Pre-Exam Tutorial

Homework Set - 01: Solution

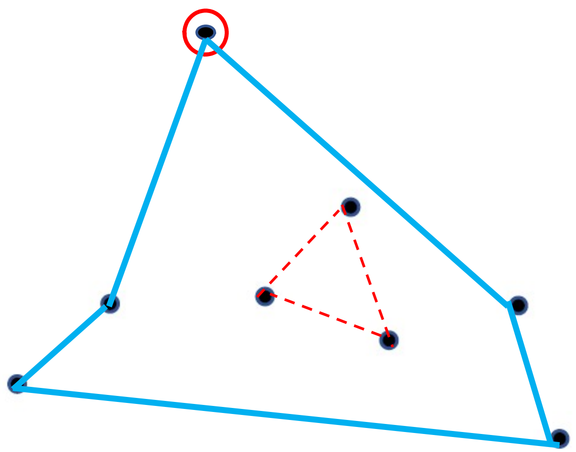
1. Given *n* points on 2D-plane, propose an algorithm to construct a simple polygon *P* with all the given points as vertices, and only those. Provide its proof of correctness and deduce its time complexity. (A simple polygon is one in which no two edges intersect each other excepting possibly at their endpoints.



Solution: Find the points with the smallest and largest *x*-value, and join them with a straight-line segment *L* (shown as dotted above). Project all points that are above *L*, on *L*. Sort these footprints and connect the original points serially in the sorted order to form the upper chain of polygon *P;* do the same for the lower chain. If one side of *L* is empty, use *L* as an edge of the polygon (as shown in the figure on the right). This can be done in $O(n \log n)$ time. There are many other solutions to this problem.

Polygonization with holes

Given n points on 2D-plane and k holes, is it always possible to polygonize the set?

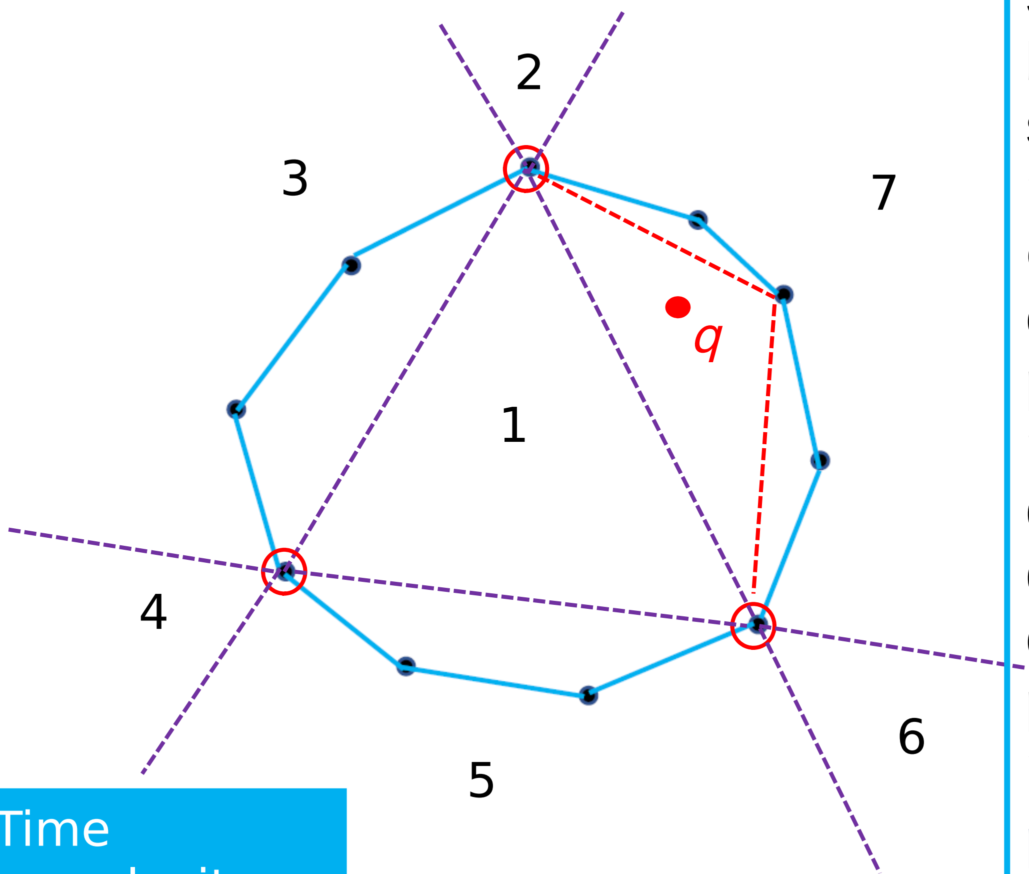


Given n points on 2D-plane and k holes, test for polygonization?

If not, where to add extra points and how to minimize their number?

Homework Set - 01: Solution

2. A convex polygon P is given as counter-clockwise ordered sequence of n vertices, in general positions, whose locations are supplied as (x, y) co-ordinates on the x-y plane. Given a query point q, propose an algorithm to determine in $O(\log n)$ time and $O(n)$ space, including pre-processing, if any, whether or not P includes q.

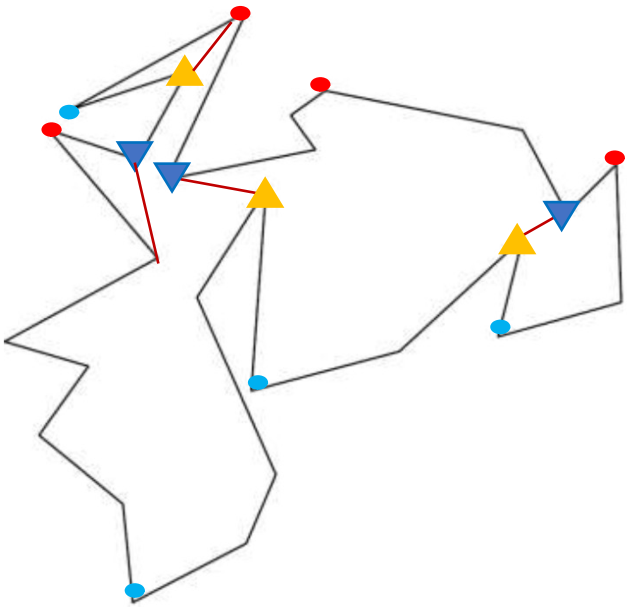


Solution: Choose three points on the boundary of P which are almost equi-spaced (divide the boundary into three nearly-equal parts - can be done in $O(1)$-time via indexing). Construct three cut-lines through these points – they partition the 2D-space into seven disjoint regions as shown. The location of the query-point q wrt these regions can be determined in $O(1)$ time via three orientation tests. Further refined partitioning can be recursed through at most $(\log_3 n)$ steps, thus giving the precise location of q in $(\log_3 n)$-time, and in $O(n)$ space.

Time complexity: $O(\log n)$

Homework Set - 02: Solution

Q1. Consider the simple polygon shown below:



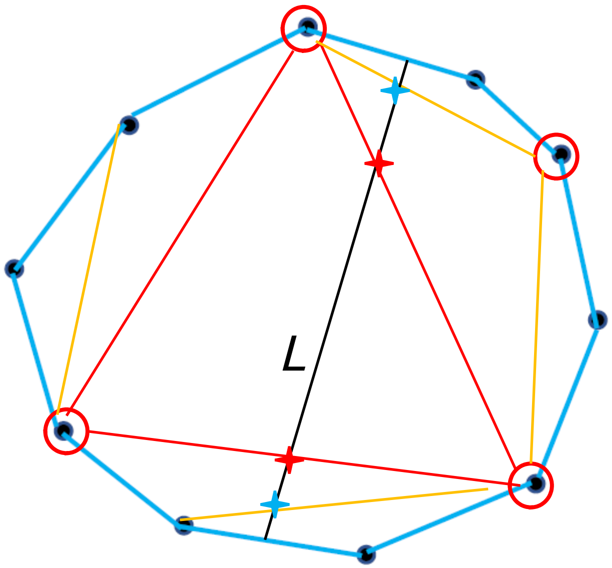
Show a partition with minimum number of *y*-monotone polygons. Justify the minimality of partition.

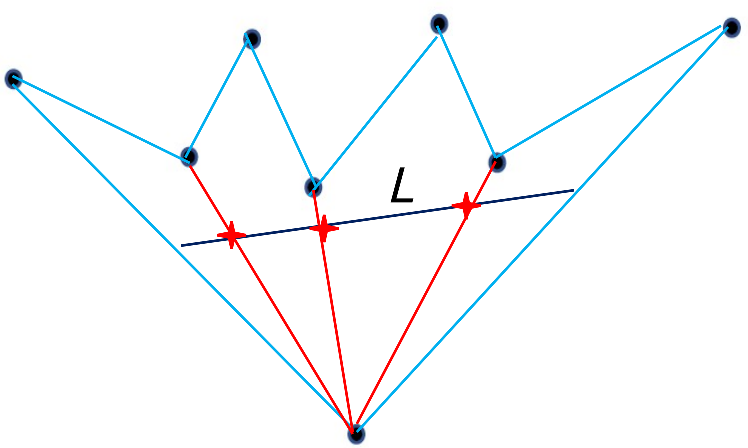
Solution: There are four top vertices (red dots), three bottom vertices (blue dots), three split vertices (yellow triangles), and three merge vertices (blue triangles). Hence, #monotone partitions is at least four. Also, one merge and one split vertex appear at the two ends of an edge of the polygon, and they cannot be matched together; neutralization reflexivity of vertices thus needs four diagonals, and hence, we obtain five monotone pieces.

Homework Set - 02: Solution

Let L be an arbitrary line segment interior to a convex polygon P with n vertices. Does there exist a triangulation such that the number of intersections of L with all diagonals become $O(\log n)$? If so, provide a method for constructing such a triangulation. Are there any polygons such that for any triangulation, such a line L

ith diagonals? If so, show an example.

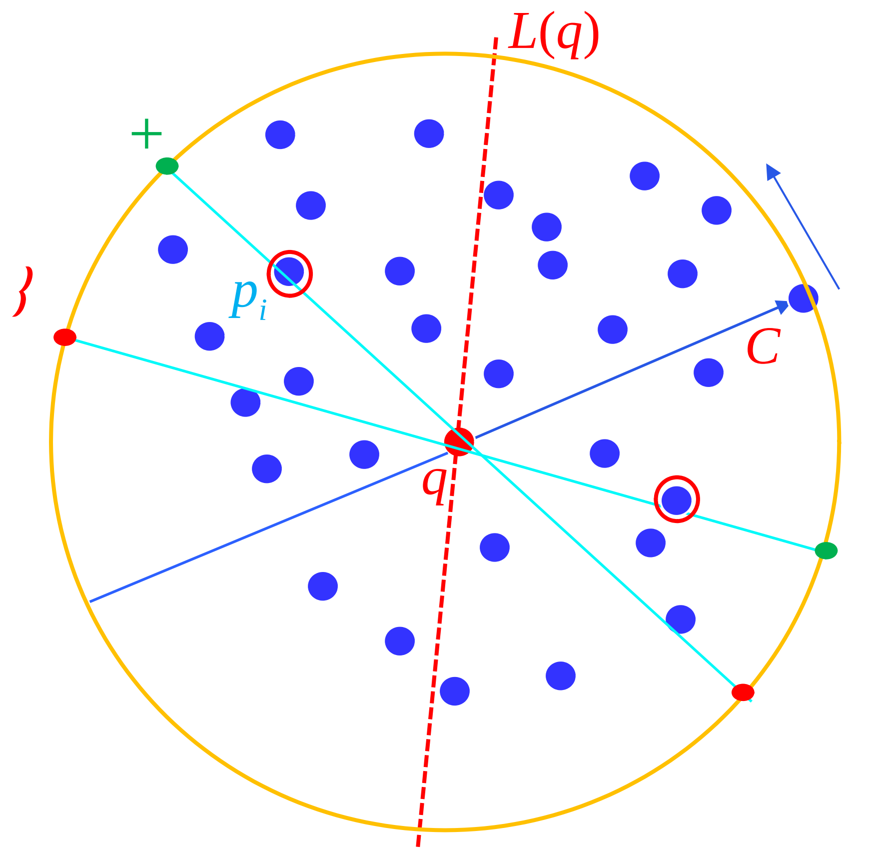




Solution: Select three vertices on the boundary of P with nearly-equal spacing. Join these diagonals. Recurse similarly through each of these peripheral components until P is fully triangulated. Any line L in P can thus cross at most 2 $\log_3 n$ diagonals. For solution to the second part, see the figure on the right; the answer is “yes”.

Homework Set - 02: Solution - Tukey Depth

Tukey depth of query point (q) in 2D:
- imagine a line L passing thru q;
- rotate L around q such that
points appearing on one side of L is *minimized* over all angles;
- Output the number including q;



Draw a circle with centre at q with the radius that is equal to the maximum radial distance of point p_i wrt q. Consider the directed diameter C passing through p_i. Radially project each point on the boundary of the circle for both ends, and label them as + or – depending on whether it appears closer or further. Let the #points on the right of C denote the initial cut-value or depth. Rotate C through ཨོཾ་ཨཱཿཧཱུྃ CCW with centre at q, and update the count depending on + or – stopping at $2n$ footprints; observe the overall least cut-value = depth.

Complexity:

Initial cut-value (#points on the right of C): $O(n)$

Initial angular sorting of $2n$ points: $O(n \log n)$

Each update: $O(1)$;

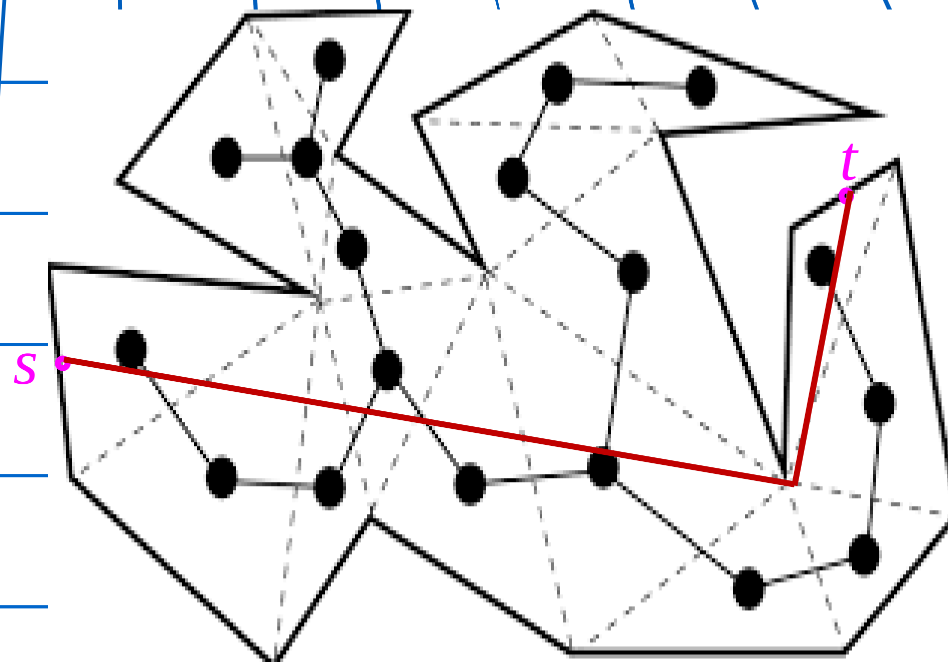
Record the running minimum over $2n$ steps;

Total: $O(n \log n)$

Homework Set - 02: Solution – ESP in a simple polygon

Solution: Let there be two shortest paths between s and *t*, and they have different branches between and . One of these branches must have a strictly convex vertex □ a shorter path between s and *t* □ contradiction!

ESP crosses only those diagonals that separate s and *t*, and each exactly once. The sequence of these triangles forms a path in the dual tree.



- The dual of a triangulated polygon is a tree

- You are given a simple polygon *P* with *n* sides and two points *s* and *t* in P, and let *T* denote a triangulation of *P*. Show that the Euclidean shortest path (ESP) between s and t is unique. Also show that the minimal set of triangles containing the ESP forms a path in the dual tree of *T*.