The Hebrew University of Jerusalem – School of Computer Science and Engineering

# Computational Geometry – Course 67842
# Final Exam – Moed A

*Date:* Thursday, March 1st, 2007
*Teacher: Prof. Leo Joskowicz*
*Duration:* 3 hours

- Please be brief, clear, and technically precise in your answers. Use drawings to illustrate your arguments and pseudo-code to describe algorithms.
- In solving the problems presented, you may assume the "general position assumption", unless it is specifically stated otherwise.
- You can assume you have subroutines for any algorithm that was given in class unless it is specifically stated otherwise.

  **Good luck**!

Choose THREE out of the following FOUR questions. You get one point just for showing up so the total is 100.

## Question 1 - Largest enclosing circle: (33%)

Let $S$ be a set of $n$ points in the plane. Define the *largest enclosing circle* as the largest circle that has all the points of $S$ in its interior or on its boundary and, as the points are in general position, exactly three points of $S$ lie on its boundary.

Let the points of $S$ on the boundary of $C$ be $s_i, s_j, s_k \in S$. It is clear that $s_i, s_j, s_k$ must be vertices of the convex hull of $S$.

1. (10%) Prove that $s_i, s_j, s_k$ are *neighbors* in the convex hull of $S$.
2. (13%) Prove that if $C$ is the largest circle incident to three consecutive vertices of the convex hull of $S$ then all the points of $S \backslash \{s_i, s_j, s_k\}$ must lie in $C$. *Hint: Thales's Theorem.*
3. (10%) Give an $O(n\log n)$ algorithm to calculate the largest enclosing circle. Show that your algorithm achieves this run time.

## Question 2 - Convex Hull merging: (33%)

You are given two arbitrary convex polygons $\mathcal{P}$ and $\mathcal{Q}$ with $n$ and $m$ vertices respectively. The polygons can be disjoint, one inside the other or the boundaries might intersect a number of times.

1. (10%) What are the smallest / largest number of vertices on the boundary of $CH(\mathcal{P} \cup \mathcal{Q})$. Give an exact expression not an asymptotic bound.

2. (13%) Give an $O(n+m)$ algorithm to compute $CH(\mathcal{P} \cup \mathcal{Q})$. Prove its correctness and analyze its complexity.

3. (10%) Assume that your answer to part 2 of this question is correct and use it to give a divide and conquer algorithm for computing the convex hull of a set of $n$ points in $O(n\log n)$ running time.

## Question 3 - External Guarding: (33%)

In class we saw how to guard the interior of an art gallery. Now we shall try to guard a building (or army base) against intruders that is we will look at the area on the out side.

More formally:

Given a polygon $\mathcal{P}$ in the plane $\Omega$ and two points $v$ and $u$, $v$ and $u$ are said to *see each other*, or *have a line of sight* if the straight line connecting them is entirely in $\mathcal{P}$ or entirely in $\Omega \backslash \mathcal{P}$ (i.e. does not properly intersect $\mathcal{P}$'s boundary). The boundary of $\mathcal{P}$ is considered to be both in $\mathcal{P}$ and in $\Omega \backslash \mathcal{P}$ so, two neighboring vertices of $\mathcal{P}$ can see each other.

We say that a guard placed at a point $v$ guards all the points that have a line of sight with $v$. A point is said to be guarded if it has a line of sight to at least one guard.

A set of guards is said to guard the union of the points guarded by the guards in the set. And an area is said to be guarded if every point in the area is guarded by at least one guard from a given set of guards.

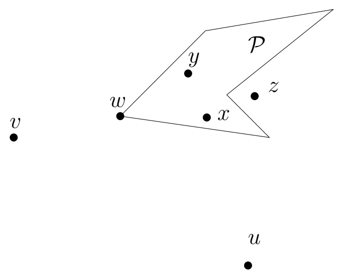



Figure 1: *In this figure $u, v, x$ and $y$ can see $w$. $x$ and $y$ can see each other (so can $v$ and $u$) none of them can see $z$.*


*(b)* *(c)*

*In this figure (b) it is illustrated that $u$ and $v$ are external guards.*
*In (c) $v$ does not guard the entire interior or the entire exterior.*

Let $G_1$ be a set of $k$ guards placed at $k$ vertices of a simple polygon $\mathcal{P}$.

1. (10%) Prove or disprove. If the exterior of $\mathcal{P}$ (i.e. $\Omega \backslash \mathcal{P}$) is guarded by $G_1$ the interior is necessarily guarded by the same set ($G_1$)?

2. (13%) Let $\mathcal{R}$ be a convex polygon with a convex hole $\mathcal{H}$. Put differently let $\mathcal{R}$ and $\mathcal{H}$ be two convex polygons so that $\mathcal{H}$ is entirely in $\mathcal{R}$. Assume that the vertices of $\mathcal{R}$ and $\mathcal{H}$ are given in CCW order $\{r_1, r_2...r_n\}$ and $\{h_1, h_2...h_m\}$ so that $r_1$ is the right most vertex of $\mathcal{R}$ and $h_1$ is the right most vertex of $\mathcal{H}$.
Give an $O(n+m)$ algorithm for triangulating $\mathcal{R} \backslash \mathcal{H}$.

3. (10%) Give a polynomial time algorithm that places the minimal number of guards (feasible by a polynomial algorithm) at vertices of $\mathcal{P}$.

   *Hint: Before you answer this you might want to clarify for yourself (not as a question to answer for the exam but as guidance) the next question: What is the smallest number of guards a sub-exponential algorithm will find (in the worst case) to guard the exterior of a simple polygon. Give an exact expression not an asymptotic bound. Answer this with respect to the next two cases, if you have the time, think give a generic example for each case:*

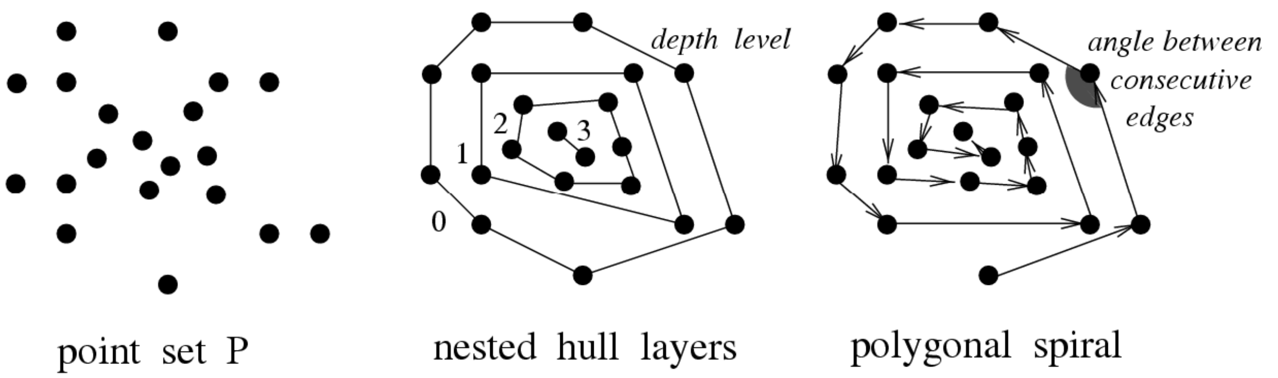
   (a) *If some guards can be placed at points exterior to $\mathcal{P}$ and some at vertices of $\mathcal{P}$.*
   (b) *If all guards are placed at vertices of $\mathcal{P}$.*

## Question 4: Nested convex hulls (33%)

Let $P$ be a set of points in the plane. Let $H(X)$ be the set of points on the convex hull of X, $conv(X)$. We define the series of nested point sets $P_i$ as follows:

- $P_0 = P$
- $P_{i+1} = P_i - H(P_i)$
  In words: the set of points of $P_i$ minus all the points of the boundary of the convex hull of $P_i$.

The sets $H(P_i)$ are called the *layers* of $P$. Point $x \in P$ has level $i$ if $x \in H(P_i)$; edge $(x, y)$ has level $i$ if it belongs to $conv(P_i)$. Assume that the points are in general position (no three points lie on a line).




1. (10%) What is the maximum number of layers of a set of $n$ planar points?
2. (10%) Design an $O(n^2 logn)$ algorithm to compute the nested convex hull layers $H(P_i)$, assign depth levels to all points in $P$ and to all edges in $conv(P_i)$
3. (13%) Design an $O(n^2)$ algorithm to compute the same as above.

# Computational Geometry – Course 67842
# Final Exam – Moed A

*Date:* Monday, July 4, 2005
*Teacher: Prof. Leo Joskowicz*
*Duration:* 3 hours

Please be brief, clear, and technically precise in your answers. Use drawings to illustrate your arguments and pseudo-code to describe algorithms.

**Good luck**!

## Question 1: general knowledge (35%)

Choose **five** out of eight of the following questions. Answer TRUE or FALSE and provide a counterexample or a **short**, one-paragraph justification. Clearly mark which five questions you want to be graded on. The right answer with the wrong explanation will be given only half the credit (no credits for wrong answers). Unless stated otherwise, assume general position and no degeneracies. If you need to make additional assumptions, write them down clearly.

1. In the art gallery problem, if a guard is placed at every *convex* vertex of a simple polygon $P$, this set of guards sees all the reflex vertices of $P$ (vertices with internal angle $> 180°$). *FALSE - counter examples exists (from homework given by Mitchell, stony brook)*

2. For every integer $n > 0$ and every axis-aligned rectangle $R$ of fixed width and height there exist $n$ planar points such that the kd-tree built for these points always has a logarithmic query time when the range is over $R$ placed somewhere in the plane. *TRUE - construct a grid so that the spacings are more than the rectangle dimensions, so that the query goes through a single branch of the tree*

3. Determining whether the convex hull of a set of $n$ points is a triangle requires $\Omega(n \log n)$ time.

   *FALSE - do three iterations of the gift wrapping algorithm and stop if original vertex not reached again: O(n)*

4. Let $P$ be a simple polygon with an even number of vertices. It is always possible to decompose $P$ by diagonals into quadrilaterals (polygons with four edges). *TRUE - every two adjacent triangles in a triangulations can be merged (there are n-2 triangles)*

5. Let $P$ be a convex polygon with $n$ vertices and let $Q$ be a simple polygon with $m$ vertices. Then the outer boundary of the Minkowski sum $P \oplus Q$ can be computed in $O((m + m) \log(n + m))$ time (the outer boundary is the the one that does not bound a hole). *FALSE - there are O(nm) vertices in the sum, even without holes*

6. Two polygons which are monotone with respect to the same line $L$ can always be separated by a single translation.
   *TRUE - proof in Toussaint's papers*

7. In the construction of the point location search structure, there always exists a sequence of insertions that causes the structure to have depth linear in the number of segments.
   *TRUE - you can construct such a sequence by sorting the segments according to right endpoints, and each insertion increases the depth by at least 1*

8. Let $S$ be a planar point set. If there exists a disc with two points $p_i, p_j \in S$ on its boundary and a third point on its interior, then $\overline{p_i p_j}$ is not an edge of the Delaunay triangulation.
   *FALSE - the lemma for Delaunay is that if there exists a disc with pi and pj on its boundary and none other are inside, then there is an edge*

## Question 2: Minimum width annulus (30%)

An annulus is the region between two concentric circles. Given a set $S$ of $n$ points, the minimum width annulus problem is finding the annulus that contains all the points of $S$ in its closure, and has the minmum width, that is the minimum difference between the outer and inner radii of the circles.

1. Let $w(p)$ be the smallest width annulus of $S$ whose center is in $p$. Describe and prove the connection between $p$ that achives the global minimum of $w(p)$ and the Nearest and Furthest point Voronoi diagrams.
   *the center is either a vertex of the nearest VD or a vertex of the furthest VD, or the intersection of an edge of the diagrams*

2. Based on your observations in the previous section, design an $O(n^2)$ time algorithm for solving the minimum width annulus problem.
   *compute overlay of diagrams, check for each vertex of overlay the width, which is the distance to the furthest site minus the distance to the closes site. Find minimum over all vertices.*

## Question 3: minimum area enclosing rectangle (35%)

Let $S$ be a set of $n$ points in the plane. We are interested in finding a rectangle, not necessarily axis aligned, that contains all the points of $S$ and has the minimum area.

1. Prove that the minimum area enclosing rectangle of $S$ has one side which is incident on at least two points of $S$.

2. Design an algorithm that computes the minimum area enclosing rectangle of $S$ and analyze its complexity.

use the rotating calipers approach, using two calipers at once. This is from the rotating calipers page:

1. Compute all four extreme points for the polygon, and call them xminP, xmaxP, yminP ymaxP. 2. Construct four lines of support for P through all four points. These determine two sets of "calipers". 3. If one (or more) lines coincide with an edge, then compute the area of the rectangle determined by the four lines, and keep as minimum. Otherwise, consider the current minimum area to be infinite. 4. Rotate the lines clockwise until one of them coincides with an edge of its polygon. 5. Compute the area of the new rectangle, and compare it to the current minimum area. Update the minimum if necessary, keeping track of the rectangle determining the minimum. 6. Repeat steps 4 and 5, until the lines have been rotated an angle greater than 90 degrees. Output the minimum area enclosing rectangle.

3. How would you modify your algorithm for the case where $S$ is a simple polygon? What would be its complexity then?
*compute the convex hull of* $S$ *with a linear time algorithm. overall complexity is linear*

The Hebrew University of Jerusalem – School of Computer Science and Engineering

# Computational Geometry – Course 67842
# Final Exam – Moed B

*Date:* Wednesday, September 21, 2005
*Teacher: Prof. Leo Joskowicz*
*Duration:* 2.5 hours

Please be brief, clear, and technically precise in your answers. Use drawings to illustrate your arguments and pseudo-code to describe algorithms.

**Good luck**!

## Question 1: general knowledge (35%)

Choose **five** out of eight of the following questions. Answer TRUE or FALSE and provide a counterexample or a **short**, one-paragraph justification. Clearly mark which five questions you want to be graded on. The right answer with the wrong explanation will be given only half the credit (no credits for wrong answers). Unless stated otherwise, assume general position and no degeneracies. If you need to make additional assumptions, write them down clearly.

1. In the art gallery problem, if a guard is placed at the midpoint of every edge of a simple polygon $P$, this set of guards sees all of $P$.

2. Every point set $S$ admits a construction of a star shaped polygon such that all the points are vertices of the polygon.

3. Let $P$ be a *convex* polygon with an even number $n > 4$ of vertices. It is always possible to decompose $P$ by diagonals into *convex* quadrilaterals (convex polygons with four edges).

4. The convex hull of all the intersection points of a set of $n$ line segments contains $\Theta(n^2)$ points in the worst case.

5. An *extreme* diagonal of a simple polygon $P$ is a diagonal whose endpoints are *both* vertices of $CH(P)$. Every non-convex simple polygon can be triangulated by non-extreme diagonals.

6. For every integer $n > 0$ there exist $n$ planar points for which the storage in a range tree is linear in size.

7. Two polygons which are monotone with respect to the same line $L$ can always be separated by a single translation.

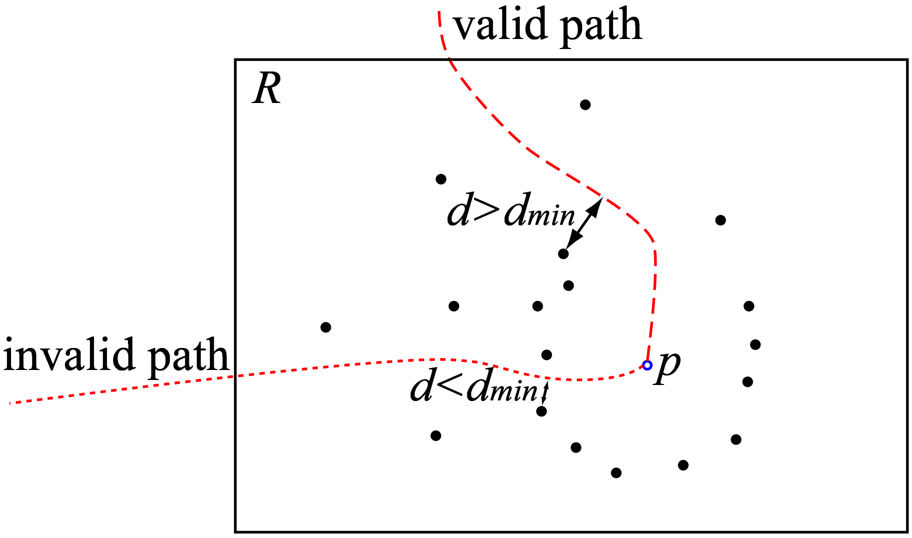
8. In Fortune's algorithm for the Voronoi diagram, the parabola defined by some site contributes at most six arcs to the beachline.

## Question 2: Path planning with clearance (30%)

Let $S$ be a set of $n$ points in the plane, and let $p$ be an additional point not in $S$. Let $R$ be a bounding rectangle that contains both $S$ and $p$ such that the minimal distance from an edge of $R$ to a point in $S$ is at least $d_{min}$. We are interested in finding a path from $p$ to the outer side of the rectangle $R$ such that at all points on the path, the minimal distance to a point in $S$ is at least $d_{min}$ (see illustration below). We say that this path has a clearance of $d_{min}$ from the set of obstacles $S$.

1. Describe an algorithm (in pseudocode) that inputs the set of obstacle points $S$, the bounding rectangle $R$, the source point $p$, and the minimum clearance distance $d_{min}$, and computes the path as describes above. The output should be the description of the path (a curve from $p$ to the outside), or a reply that no such path exists. Analyze the time and space complexity of your algorithm.

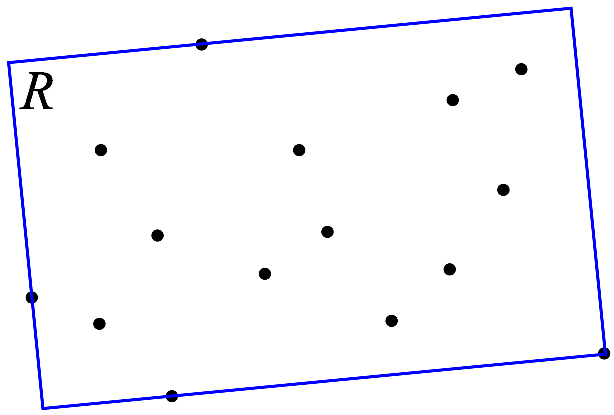
2. Briefly describe how to extend your algorithm to solve the problem where $S$ is a set of convex polygonal obstacles. You may use known data structures and algorithms as black boxes.



## Question 3: Minimum area enclosing rectangle (35%)

Let $S$ be a set of $n$ points in the plane with no three collinear points. We are interested in finding a rectangle $R$, not necessarily axis aligned, that contains all the points of $S$ and has the minimum area.

1. Prove that the minimum area enclosing rectangle of $S$ has one side which is incident on at least two points of $S$.
2. Describe an algorithm (data structures and pseudocode) that inputs the set of points $S$ and computes the minimum area enclosing rectangle $R$. Analyze the algorithm's space and time complexity.
3. How would you modify your algorithm for the case where $S$ is a simple polygon? What would be its complexity then?



67.599
תשס"ב / א'



67599

# Computational Geometry – Course ~~67842~~

## Final Exam – Moed A

גיאומטריה חישובית

*Date:* Sunday, June 30, 2002
*Teacher: Prof. Leo Joskowicz*
*Duration:* 3 hours

Please be brief, clear, and technically precise in your answers. Use drawings to illustrate your arguments and pseudo-code to describe algorithms.

**Good luck!**

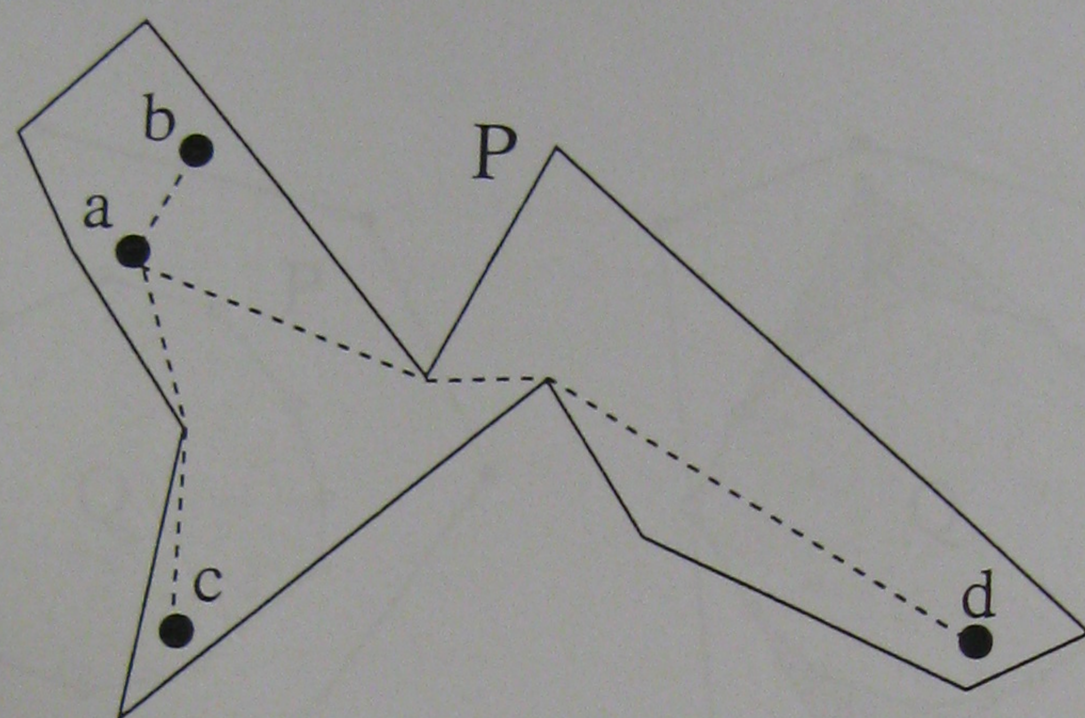
### Question 1: general knowledge (35%)

Choose **five** out of eight of the following questions. Answer TRUE or FALSE and provide a counterexample or a **short**, one-paragraph justification. Clearly mark which five questions you want to be graded on. The right answer with the wrong explanation will be given only half the credit (no credits for wrong answers). Unless stated otherwise, assume general position and no degeneracies. If you need to make additional assumptions, write them down clearly.

1. Testing if a closed chain of $n$ line segments is the boundary of a simple polygon can be done in $O(n \log n)$ time.
2. A simple polygon $P = \{p_1, ..., p_n\}$ is star-shaped iff every triple of **convex** vertices $(p_i, p_j, p_k)$, not necessarily sequential, is visible by a point $q_{ijk}$ inside $P$.
3. The maximum number of neighbors of a Voronoi cell is six.
4. The visibility region of a point inside a simple polygon is always a simple polygon (assume strict visibility: two points inside a polygon are visible iff there is a straight line that does not intersect or overlap any polygon edge or vertex).
5. The convex hull of a set of $n$ points in 3D can be computed from the Voronoi diagram of their projection in the $xy$ plane in $O(n)$ time.
6. The smallest axis-aligned rectangle enclosing all intersection points of $n$ lines in general position (no parallel lines, no more than two lines intersect at a point) can be computed in $O(n \log n)$.
7. Axis-aligned orthogonal range queries in the plane are always more efficiently answered with range trees than with kd-trees.
8. The visibility graph between a set of polygons in the plane whose total number of vertices is $n$ can be computed in time $O(n \log n)$.

## Question 2: link distance of a simple polygon (30%)

Let $P = \{p_1, ...p_n\}$ be a simple polygon with $n$ vertices. We define the *link distance* between two points $a$ and $b$ inside $P$ to be the minimum number of straight line segments necessary to go from $a$ to $b$ without exiting $P$ (touching the vertices and edges of $P$ is allowed):




*link_distance(a,b) = 1*

*link_distance(a,c) = 2*

*link_distance(a,d) = 3*

*link_distance(P) = 3*

We define the *link distance of a polygon* $P$ to be the maximum of the link distances between any two points inside $P$.

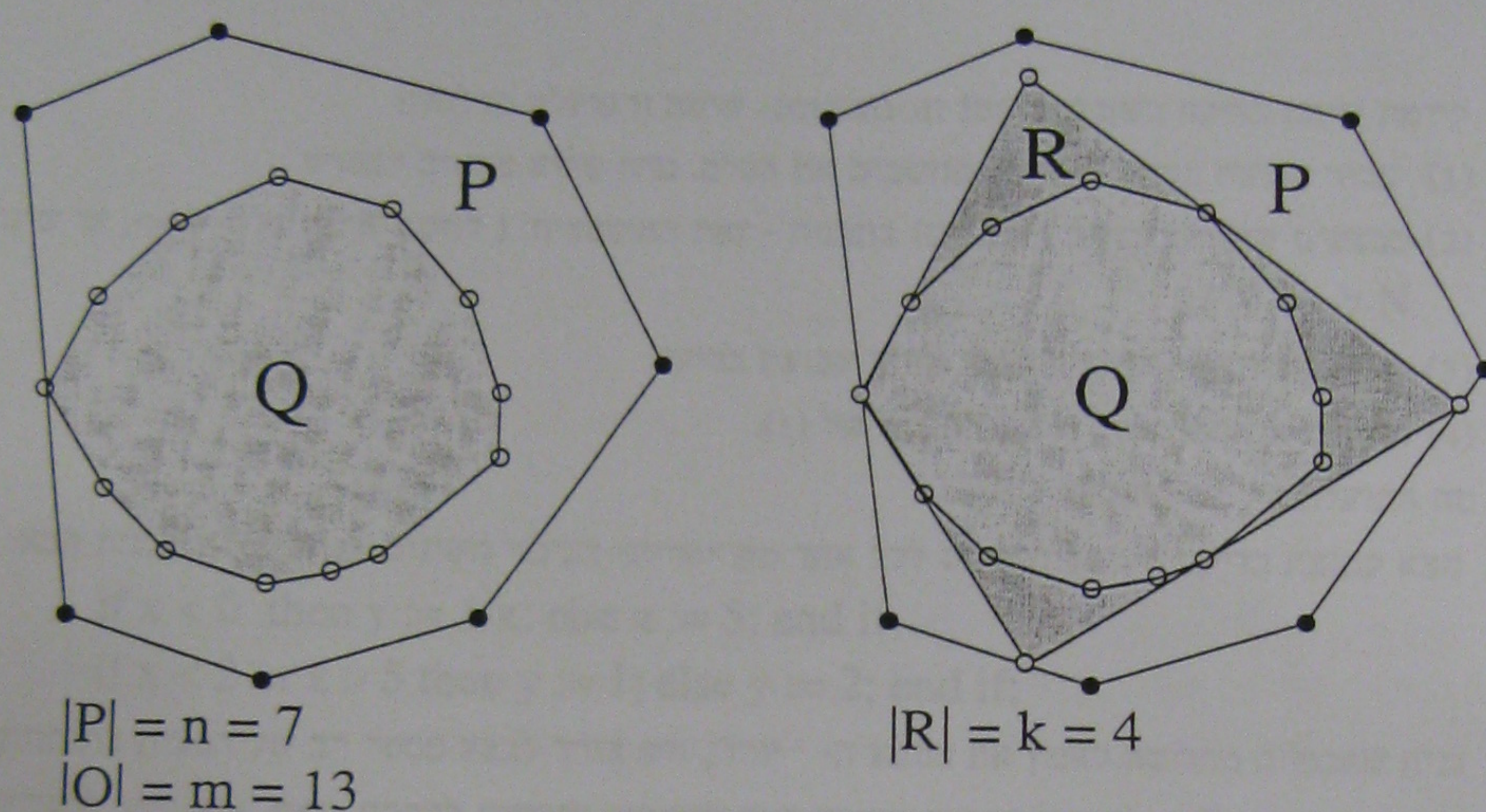
1. Give an upper and lower bound for the link distance of a polygon with $n$ vertices as a function of $n$ (the actual number, not just big-oh). Illustrate each case with an example.
2. Show that the link distance of a polygon $P$ is realized by a pair of vertices $(p_i, p_j)$ of $P$.
3. Describe an efficient data structure and algorithm to compute the link distance of a polygon $P$. What is the time and space complexity of your algorithm?

   Try to achieve a time complexity which is less than $O(n^2 \log n)$ in the number of vertices in $P$. You will get extra bonus for a lower complexity.

## Question 3: Smallest nested convex polygon (35%)

Let $P = \{p_1, ..., p_n\}$ and $Q = \{q_1, ..., q_m\}$ be two convex polygon with $n$ and $m$ vertices respectively so that polygon $Q$ is contained in polygon $P$, $Q \subset P$.

We want to find another polygon $R = \{r_1, , , , , r_k\}$, called the *smallest nested convex polygon* between $P$ and $Q$ so that $R$ is nested between $P$ and $Q$, $Q \subset R \subset P$ and $R$ has the minimum number of vertices $k$:




Note that we are only interested in a minimum number of vertices for $R$, not in a minimum area for it. Note also that there might be more than one such polygon, so we are only interested in one of them. Polygon vertices can be on edges or in the interior of $P$, and edges can be parallel to each other.

1. Give an upper and lower bound for the number $k$ of vertices of $R$ as a function of $n$ and $m$ (the actual number, not just big-oh). Illustrate each case with an example.
2. Show that there is a smallest nested convex polygon $R$ with the following two properties:
   (a) the vertices of $R$ are either vertices of $P$ or $Q$ or are on edges of $P$.
   (b) the edges of $R$ overlap edges of $Q$ or include a vertex of $Q$.
3. Describe an efficient algorithm to compute the smallest nested convex polygon $R$ between two nested convex polygons $P$ and $Q$. What is the time and space complexity of your algorithm?

   Try to achieve a time complexity which is no higher than quadratic in the number of edges of $P$ and $Q$.

   *Hint:* One possible solution is to translate the problem to the dual plane.

67.599
תשס"א/א'

The Hebrew University of Jerusalem – School of Computer Science and Engineering

ג'יאומטריה חישובית

# Computational Geometry – Course 67599

ג'יאומטריה
[illegible]

## Final Exam – Moed A

*Date:* Tuesday, July 3rd, 2001
*Teacher: Prof. Leo Joskowicz*
*Duration:* 3 hours

Please be brief, clear, and technically precise in your answers. Use drawings to illustrate your arguments and pseudo-code to describe algorithms.

Good luck!

### Question 1: general knowledge (35%)

Choose **five** out of eight of the following questions. Answer TRUE or FALSE and provide a counterexample or a **short**, one-paragraph justification. Clearly mark which five questions you want to be graded on. The right answer with the wrong explanation will be given only half the credit (no credits for wrong answers). Unless stated otherwise, assume general position and no degeneracies. If you need to make additional assumptions, write them down clearly.

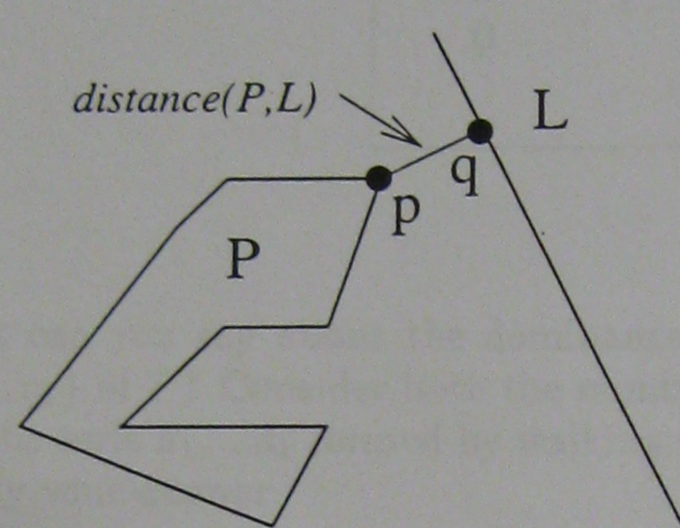
1. A simple polygon $P$ is monotone if it has no interior cusps (an interior cusp is a concave vertex whose neighbors are above or below it).
2. The number of edges of a Voronoi cell averaged over all cells is equal or smaller to six.
3. In three-dimensions, the the size of the Delaunay triangulation (into tetrahedra) of a set of $n$ points is linear in $n$.
4. The dual graph of **any** triangulation of a monotone polygon that uses the original vertices only is always a chain, that is, all but two nodes in this graph are of degree two.
5. Determining if a simple polygon $P$ with $n$ vertices is star-shaped requires at least $O(n \log n)$ time.
6. Reporting the **number of points** inside antwo axis-parallel rectangle using a kd-tree structure storing $n$ points can be performed in $O(\sqrt{n})$
7. An Euler tour of the vertices of an arrangement $A(L)$ of $n$ lines $L$ in general position in the plane can be constructed in $O(n^2)$ (an Euler tour of a graph is a path that visits each vertex exactly once).
8. Two star-shaped polygons can always be separated by a single translation of one of them in a fixed direction.

67. 599
16/6'cPN

## Question 2: Line-polygon distance queries (30%)

Let $P$ be a simple polygon with $n$ vertices, and $L$ be a line in the plane which does not intersect it. The distance between the polygon $P$ and the line $L$ is defined as the minimum distance between a point in $P$ and a point in $L$:

$$distance(P, L) = \min\{|p - q| : p \in P \wedge q \in L\}$$




In the following, assume tha the line $L$ does not intersect any of the edges of the polygon $P$, and that the line is not parallel to any of the polygon edges.

1. Show that the distance between the polygon $P$ and a line $L$ is the distance between one of the vertices in the convex hull of $P$ and the line $L$.

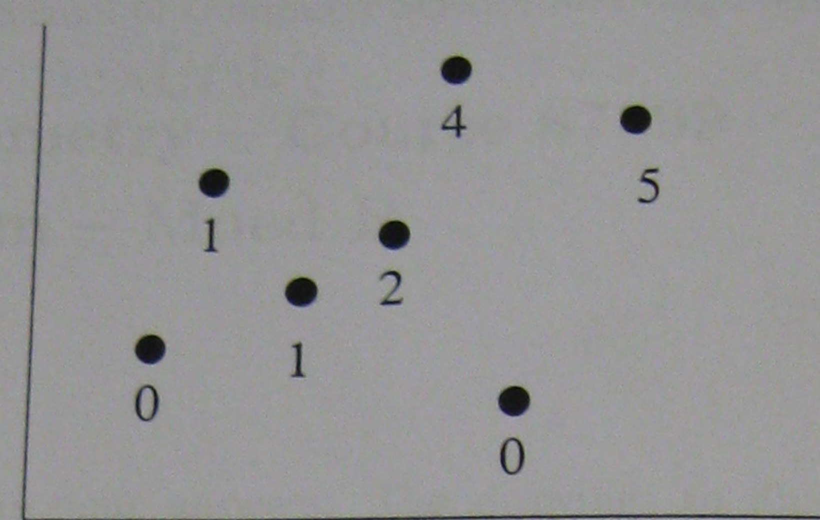
2. Design a data structure, preprocessing algorithm, and query algorithm to answer a series of line-polygon queries for polygon $P$ and lines $L_1, \ldots L_m$. The storage should $O(n)$, and the query time $O(\log n)$.

*Hint:* one possibility is to look for a method for doing binary search of the hull edges based on the orientation of the line and the convexity of the hull.

07.544
16/6 oPR

### Question 3: dominance numbers of a set of points (35%)

Let $P$ be a set of $n$ points in the plane. We say that point $p = (p_x, p_y)$ **dominates** point $q = (q$
iff $q_x \leq p_x$ and $q_y \leq p_y$, where $p$ and $q$ are distinct points (when $q_x \geq p_x$ or $q_y \geq p_y$, p does
dominate q). The **dominance number** of a point $p$ in $P$ is the number of points that it domin
Point $p$ is said to be a maximum if no point in $P$ dominates it, and a minimum if all points domi
it.




1. What can you say about the dominance numbers $d_i$ of the vertices in the convex hull $\{q_1, .., q_k\}$ of $P$? Consider both the numbers $d_i$ themselves (minimum, maximum, repetiti and the serie $d_1, ...d_k$ formed by walking counterclockwise on $C$. Justify your answer.
2. Describe an efficient data structure and algorithm to compute the dominance numbers points in $P$. Your algorithm should take time less than $O(n^2)$. What is the time and s complexity of your algorithm?
3. Develop an efficient algorithm to delete a point in $P$ and update the dominance number remaining points in $P$. Your algorithm should take less time than computing the domin number of the new set $P - \{p\}$. There is no need to perform maintenance operations su tree balancing after the deletion of the point. What is the worst-case time complexity of algorithm?

67 599
תשס״א/ב

The Hebrew University of Jerusalem – School of Computer Science and Engineering

גיאומטריה חישובית

גיאומטריה חישובית

# Computational Geometry – Course 67599

## Final Exam – Moed B

*Date:* August 2001
*Teacher: Prof. Leo Joskowicz*
*Duration:* 3 hours

Please be brief, clear, and technically precise in your answers. Use drawings to illustrate your arguments and pseudo-code to describe algorithms.

**Good luck!**

### Question 1: general knowledge (35%)

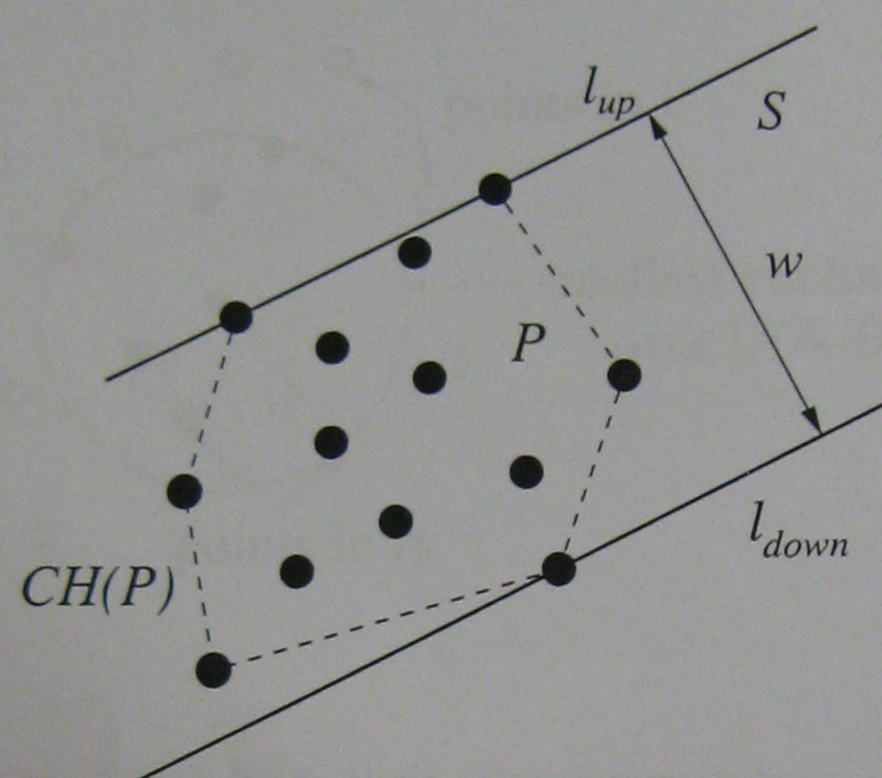
Choose **five** out of eight of the following questions. Answer TRUE or FALSE and provide a counterexample or a **short**, one-paragraph justification. Clearly mark which five questions you want to be graded on. The right answer with the wrong explanation will be given only half the credit (no credits for wrong answers). Unless stated otherwise, assume general position and no degeneracies. If you need to make additional assumptions, write them down clearly.

1. The gift wrapping algorithm for computing the convex hull of a set of points in the plane is always more efficient than the Graham scan algorithm when more than half of the points in $P$ are hull vertices.
2. Star-shaped polygons can be triangulated in time linear in their number of vertices
3. The number of monotone subdivision chains of a simple polygon $P$ with respect to a line $L$ is proportional to the number of interior cusps of $P$ with respect to a line orthogonal to $L$ (an interior cusp is a concave vertex whose neighbors are above or below it).
4. The visibility graph of $n$ non-intersecting segments in the plane can be computed in $O(n^2)$ time.
5. We can determine in $O(n^2)$ time if a set of $n$ points in the plane has three colinear points.
6. Answering planar orthogonal range queries using range trees is always equally or more efficient than using uniform quad trees for **any** type of data in terms of storage and query processing time.
7. Testing if a given triangulation of points in the plane is a Delaunay triangulation can be done in time linear in the number of points.
8. The size of the Minkowski sum of two simple polygons with $n$ and $m$ vertices respectively is $O(nm)$.

67.599
תשס'א/ב'

## Question 2: the width of a point set (30%)

Let $P$ be a set of $n$ points in the plane (no three points on a line, no four points on a circle). A strip $S$ is the region of the plane contained between two parallel lines, $l_{up}$ and $l_{down}$. The width $w$ of a strip $S$ is the distance between its two defining lines. The width $w$ of a point set $P$ is the minimum width of the strip $S$ containing all points in $P$.

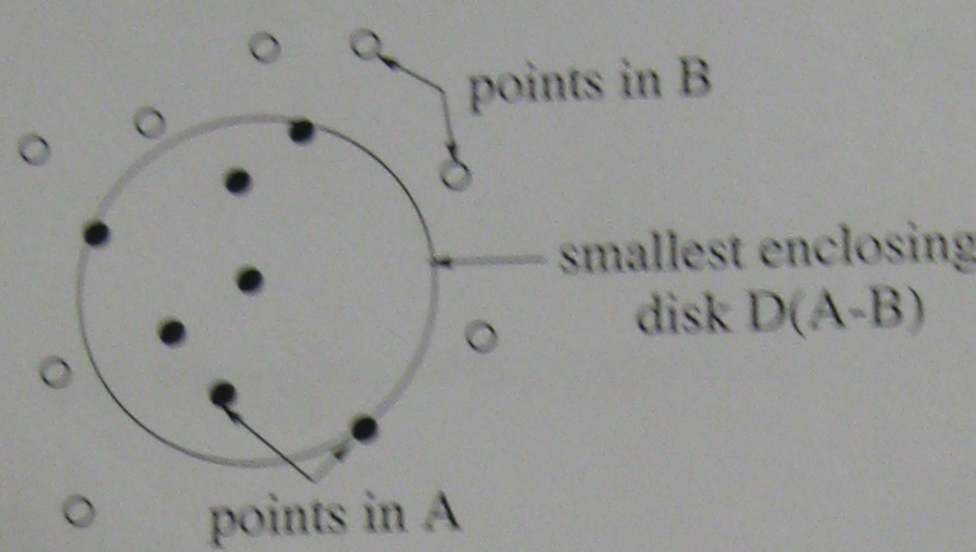



1. Show that the lines of strip $S$ with width $w$ on $P$ are defined by an edge and a vertex, or by two edges, of the convex hull $CH(P)$ of $P$.

2. Describe an $O(n \log n)$ algorithm to compute the width of a set of points $P$.

## Question 3: Smallest enclosing circle of A excluding B (35%)

Let $A$ and $B$ be two sets of distinct planar points of size $n$ and $m$ respectively. We are interested in finding (if it exists) the smallest enclosing disk $D(A-B)$ that contains every point of $A$ and excludes every point in $B$. You can assume that the points are in general position, that is, that no four points in $A$, $B$, or $A \cup B$ lie on a circle).




1. Define the conditions on $A$ and $B$ (or their related structures – convex hull, Voronoi Diagram, etc) that guarantee the existence of a solution.
2. Develop an efficient algorithm to determine if there exist a smallest enclosing disk for two disjoint sets $A$ and $B$, $D(A-B)$, and if so, to compute the center and radius of this disk. What is its time and space complexity of your algorithm?

67599

האוניברסיטה העברית בירושלים
המכון למדעי המחשב

<u>מבחן סופי בגיאומטריה חישובית</u>
קורס מסי 67599

מועד בי תשנ"ח
המורה: ד"ר לאו יוסקוביץ

תאריך: 31.8.97
זמן: שעתיים וחצי

אנא נסחו את תשובותיכם בקיצור, בבהירות ובדיוק. השתמשו בשרטוטים כדי להמחיש את טענותיכם. בארו אלגוריתמים בפסאודו-קוד.

## <u>שאלה 1 : מושגים כלליים</u> (40%)

בחרו 5 מתוך 8 השאלות שלהלן. ענו "נכון" או "לא נכון" והצדיקו את תשובתכם ע"י טיעון קצר (<u>פיסקה אחת</u>) או דוגמה נגדית. תיבדקנה רק חמש התשובות הראשונות שתופענה.

### <u>שאלה 1</u>

1. ניתן להחליט אם שתי מצולעים כוכביים (star-shaped polygons) נחתכים בזמן לינארי במספר קודקודיהם.

2. ניתן לתאר פאון פשוט (תלת מימדי) ע"י גרף שכנויות שבו הקודקודים הם הפאות והקשתות הן היחסי שכנות ביניהם.

3. ניתן לחלק כל מצולע פשוט בעל מספר זוגי של קודקודים n לשני מצולעים פשוטים עם n/2 קודקודים כל אחד ע"י אלכסון (צלע בין שני קודקודים שלא נחתך עם אף צלע אחרת במצולע).

4. כל עץ בינארי ניתן ליישום ע"י שלוש דואלי (dual triangulation) של מצולע.

5. ניתן לחשב את הקמור של שני פאונים נפרדים בזמן לינארי במספר פאותיהם.

6. ניתן לחשב את הקמור של קבוצת נקודות במישור מדיאגמת וורוני (Voronoi Diagram) בזמן לינארי במספר הנקודות.

7. ניתן להשמיט קו בהסדר (arrangement) ולעדכן את מבנה הנתונים המתאר את ההסדר בזמן לינארי במספר הקווים.

8. ניתן לחשב את המרחק הקטן והגדול ביותר בין שתי נקודות בקבוצה נתונה בזמן לינארי בהינתן שלוש Delaunay שלהם.

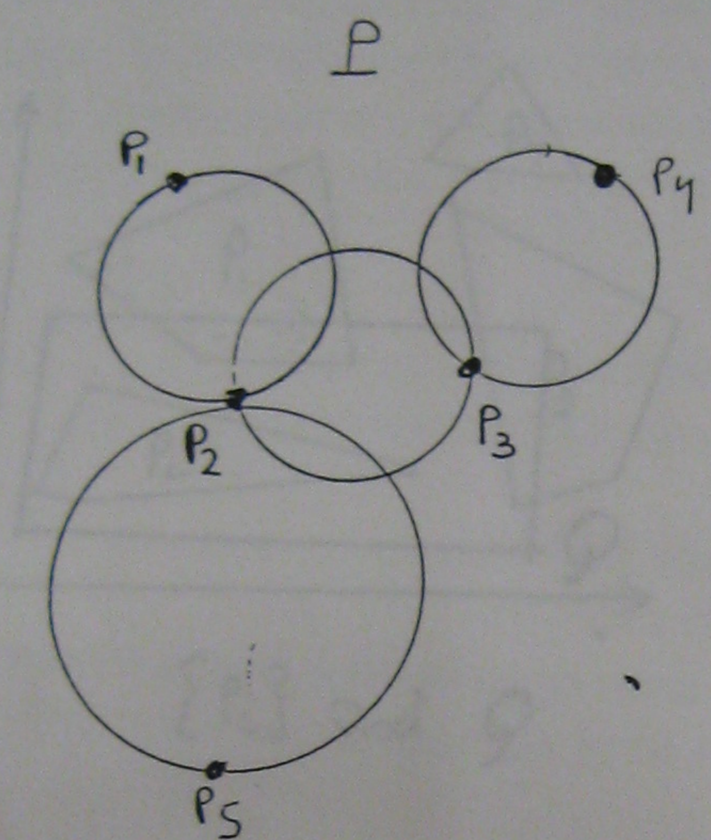
2/...

67.599
תשנ"ז ב

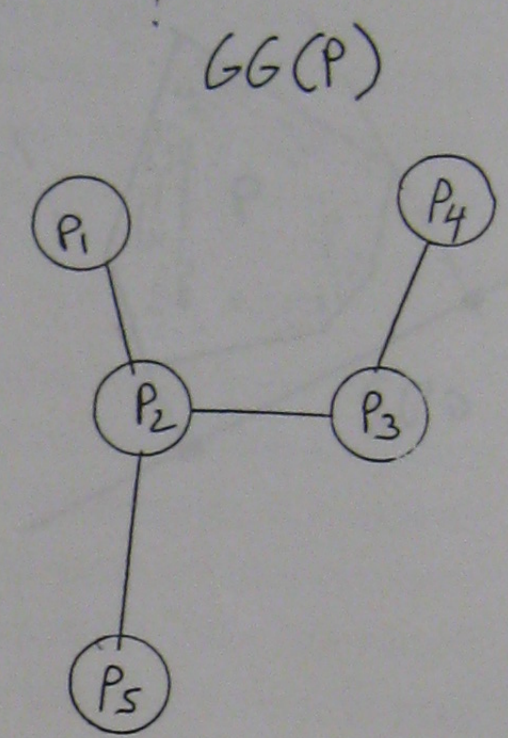
שאלה 2 - גרף גבריאל (30%)

גרף גבריאל GG(P) של קבוצה P של n נקודות במישור מוגדר כדלקמן:

- הקודקודים של GG(P) הן הנקודות P
- שני קודקודים p ו- q מחוברים בצלע pq אם ורק אם המעגל העובר דרך p ו- q שקוטרו המרחק בין p ו- q לא מכיל נקודה אחרת ב- P.




Original set of Points




Gabriel Graph

1. הוכח שכל צלע $pq \in GG(P)$ גם צלע בגרף של השלוש Delaunay של P, DT(P), כלומר:

$$pq \in GG(P) \Rightarrow pq \in DT(P)$$

האם ההיפך נכון? נמק.

$$pq \in DT(P) \Rightarrow pq \in GG(P)$$

2. האם הגרף GG(P) תמיד קשיר? נמק.

3. תאר אלגוריתם המחשב את GG(P) בזמן $O(n \log n)$.

3/...

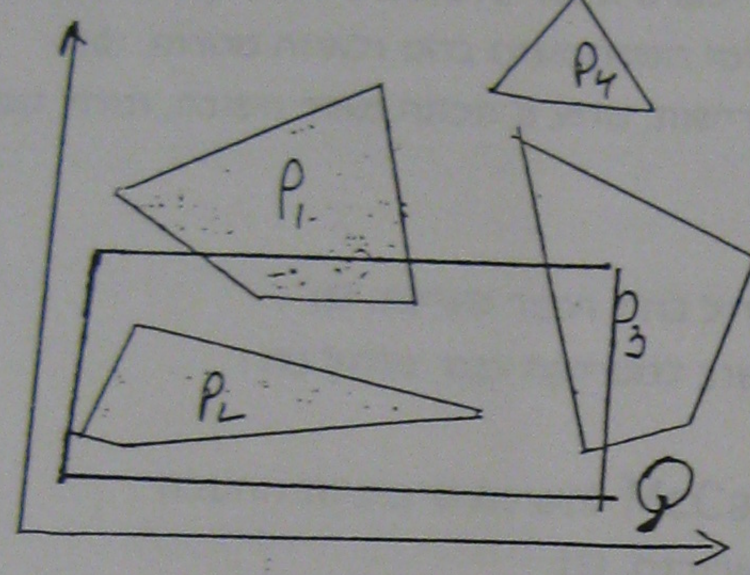
67.599
תשנ"ג ב'

<u>שאלה 3 :  שאילתות טווח על מצולעים (Range queries over polygons)</u> (30%)

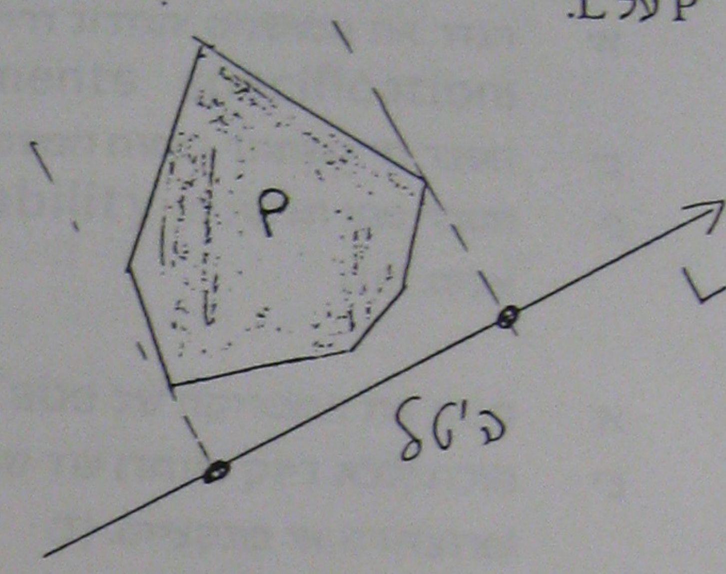
תהי $\{P_i\}$ קבוצה של מצולעים קמורים שלא נחתכים. מספר הקודקודים בכל המצולעים יחד הינו n.

אנו מעוניינים לפתח אלגוריתמים המסוגלים להשיב בצורה יעילה, לאחר חישובי הכנה מקדימים על ה- $P_i$, עם אלו מה- $P_i$ נחתך במצולע קמור נתון Q.

נגדיר את ההיטל של מצולע קמור P על ציר L כאינטרוול הקטן ביותר המכיל את היטל הנקודות של P על L.




{Pi} and Q




1. הוכח ששני מצולעים קמורים נפרדים (disjoint) אם ורק אם קיים ציר L כך שההיטלים של שני המצולעים על L לא נחתכים. לציר L כזה נקרא <u>ציר הפרדה</u>.

2. נניח עכשיו שהמצולעים $P_i$ והשאילתה Q הם מלבנים שצלעותיהם מקבילים לצירים x ו- y (axis-aligned rectangles). תארו מבנה נתונים יעיל לאחסון המצולעים $P_i$, ואלגוריתם יעיל עונה על שאילתה Q. מה הסיבוכיות של בנית מבנה הנתונים ומה סיבוכיות הטיפול בשאילתה?

3. כנ"ל עבור המקרה שהמצולעים $P_i$ קמורים והשאילתה Q הינה מלבן שצלעותיו מקבילים לצירים.

<u>רמז</u> : קרבו את המצולעים הקמורים בעזרת מלבנים.

<u>בונוס</u> : תארו אלגוריתם יעיל למציאת ציר מפריד בין שני מצולעים קמורים נפרדים.

**בהצלחה!**