Competitive Programming

GEOMETRIC ALGORITHMS

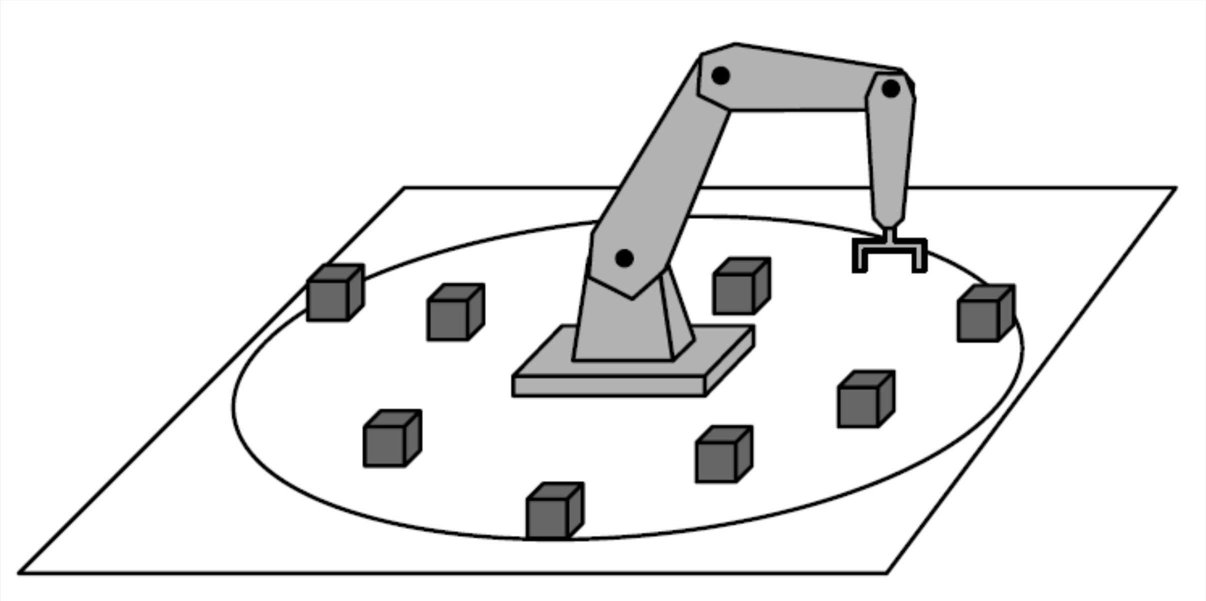
Hamid Zarrabi-Zadeh

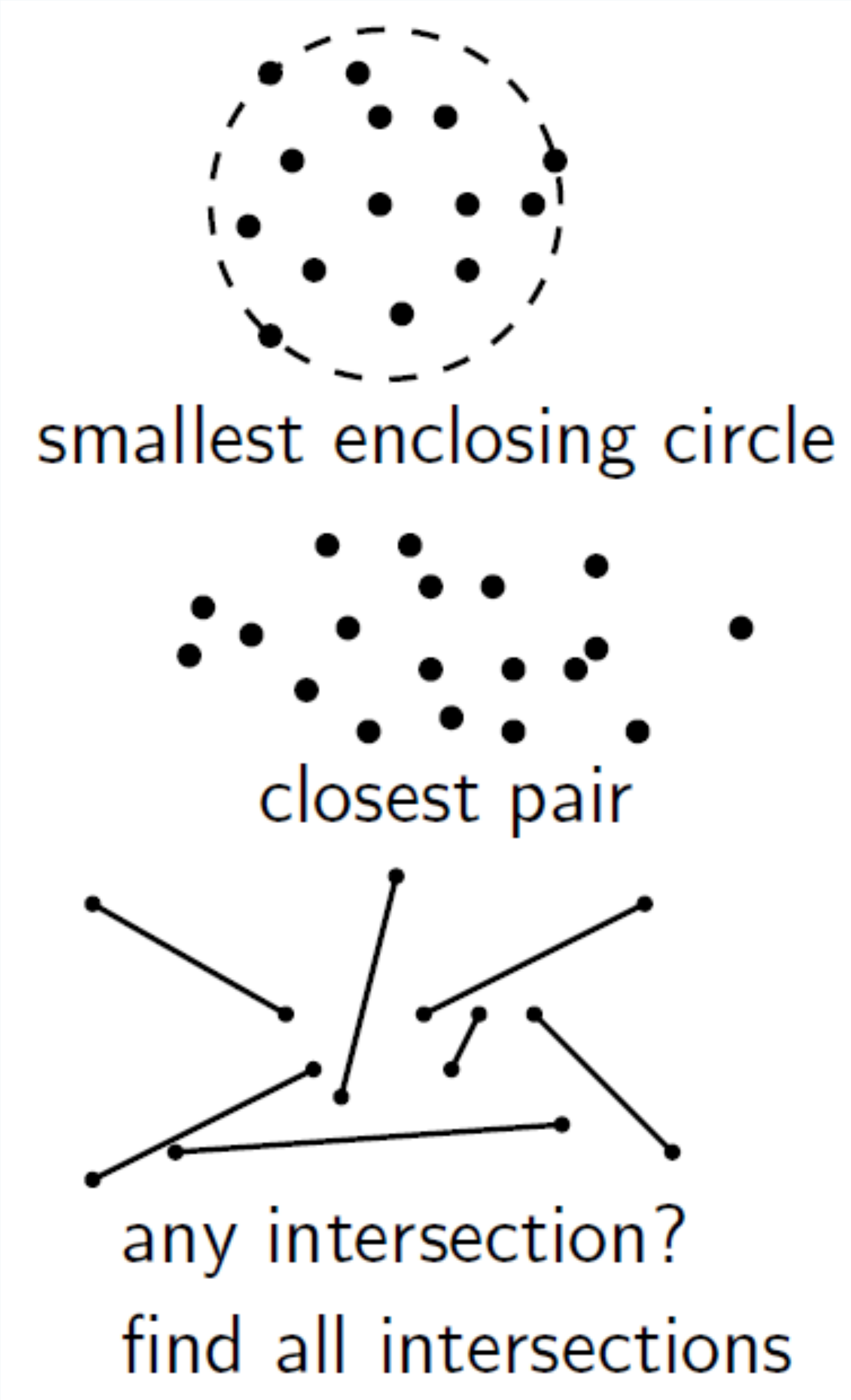
Sharif University of Technology

Spring 2024

Representative problems

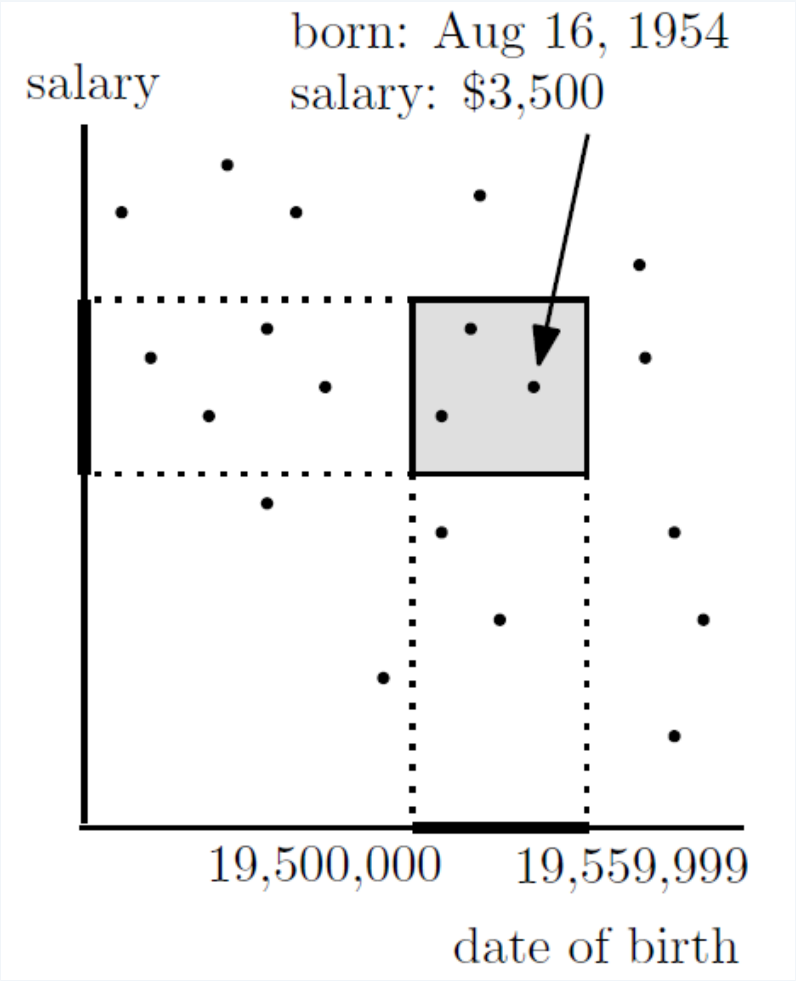
- Smallest enclosing circle
- Closest pair
- Segment intersections

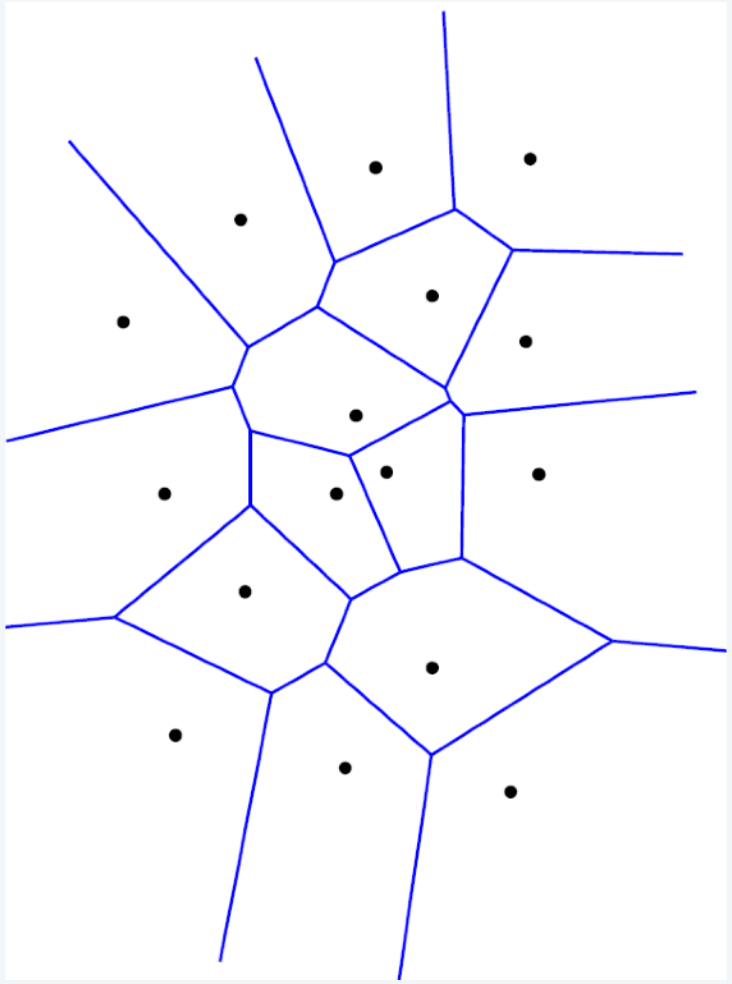


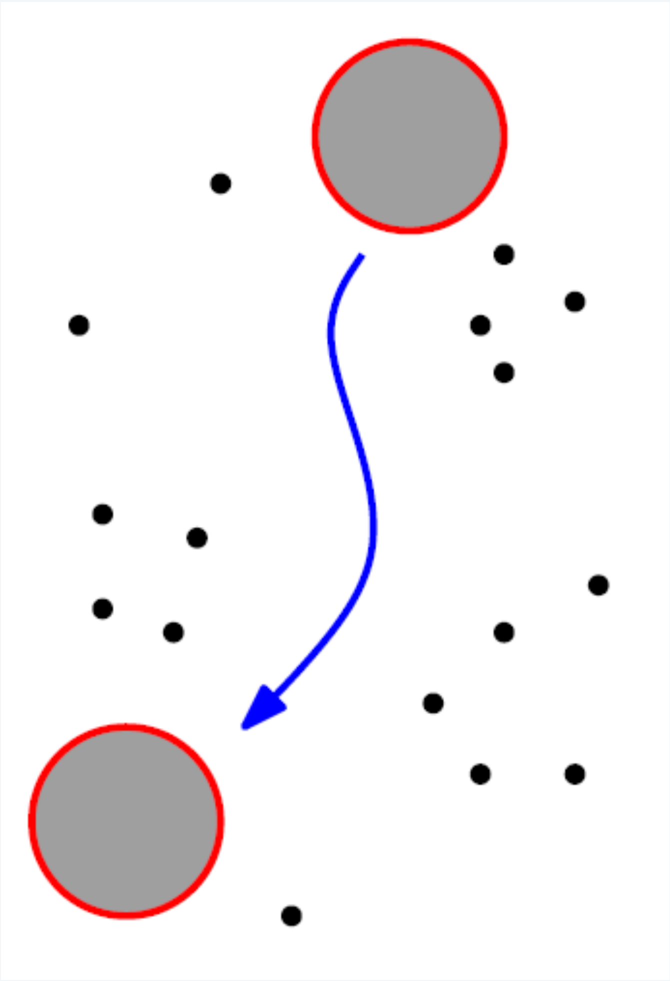


Representative problems (cont'd)

- Range searching
- Nearest neighbor search
- Motion planning

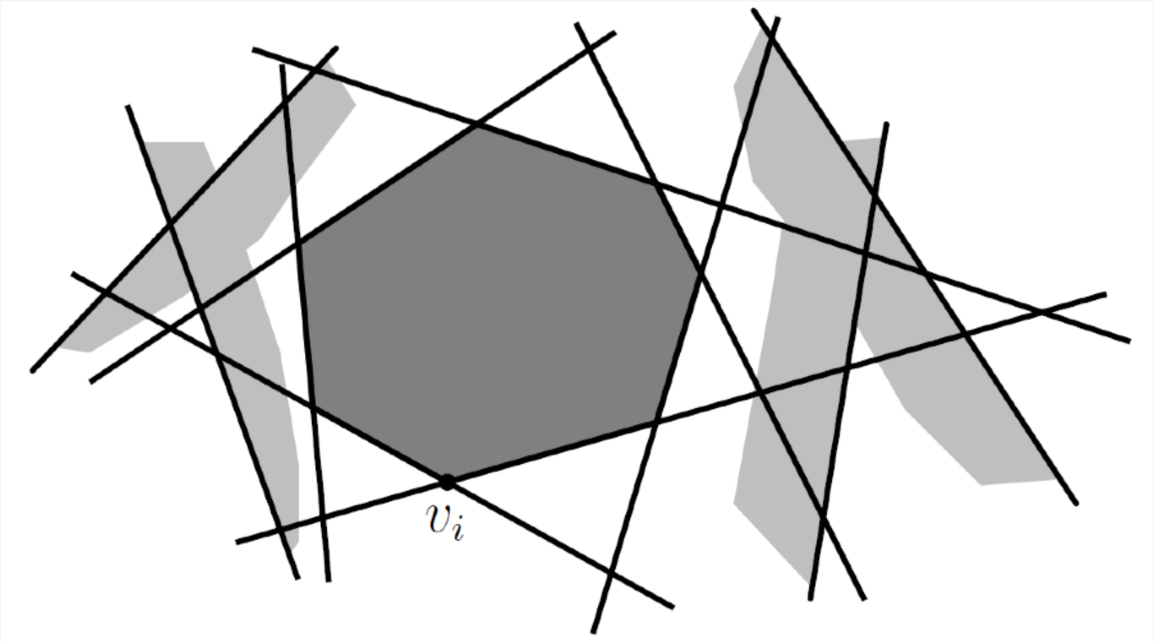


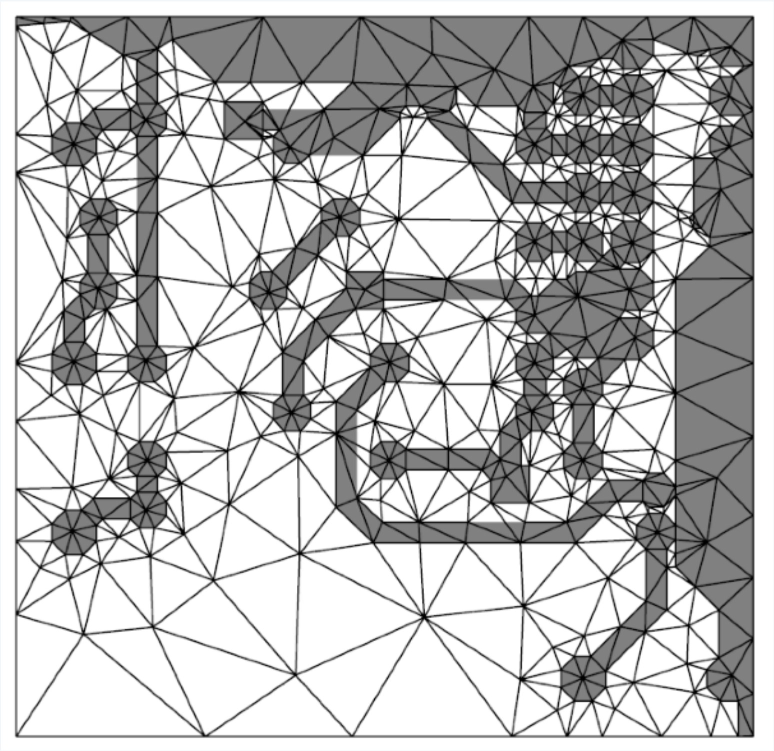


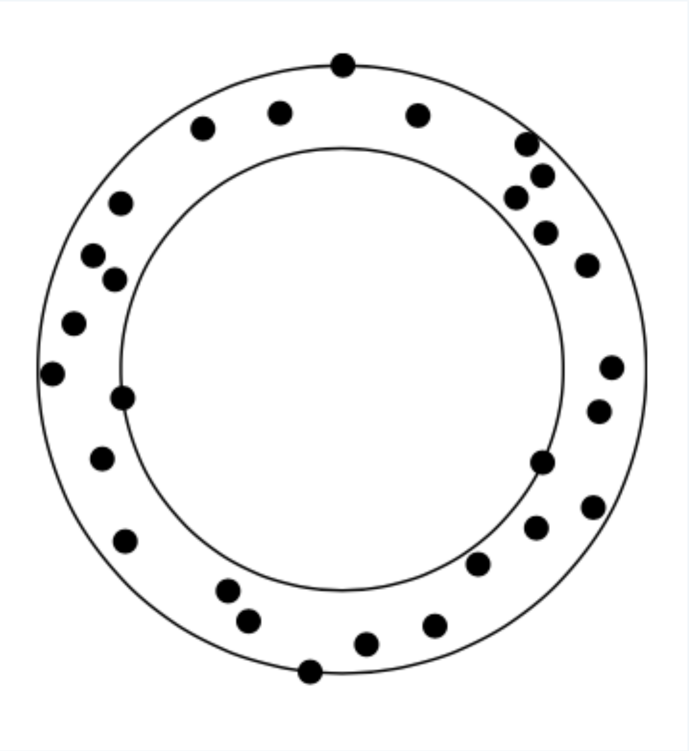


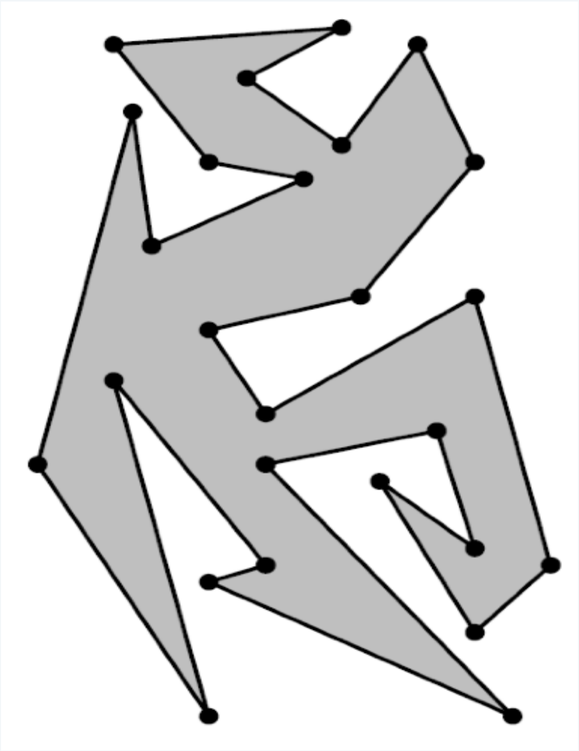
Geometric objects in 2D

- Point
- Line / Segment
- Circle / Disk
- Polygon / Triangle
- Polygonal Chain
- Half-plane
- ...

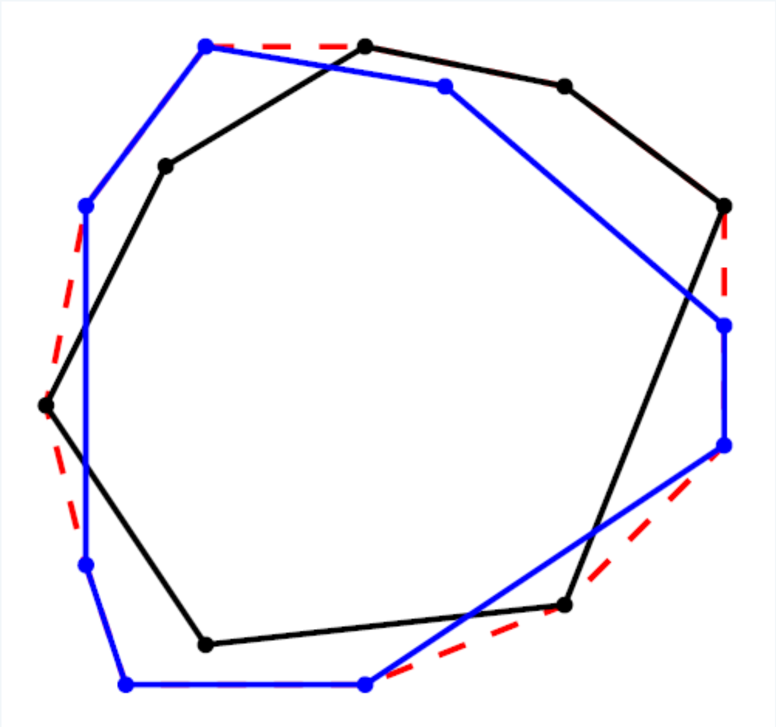








Geometry problems



In programming contests:

- Significant impact on the result
- Tricky to implement!

Implementation Issues:

- Degenerate cases
- Precision

General advices:

- Avoid divisions as much as possible
  - e.g., Instead of `if(a/b == c)`, write `if(a == b*c)`
- Avoid using floating-point number whenever you can
- Use long long and double instead of int and float
- Use epsilon in equality tests
  - e.g., Instead of `if(x == 0)`, write `if(abs(x) < EPS)`

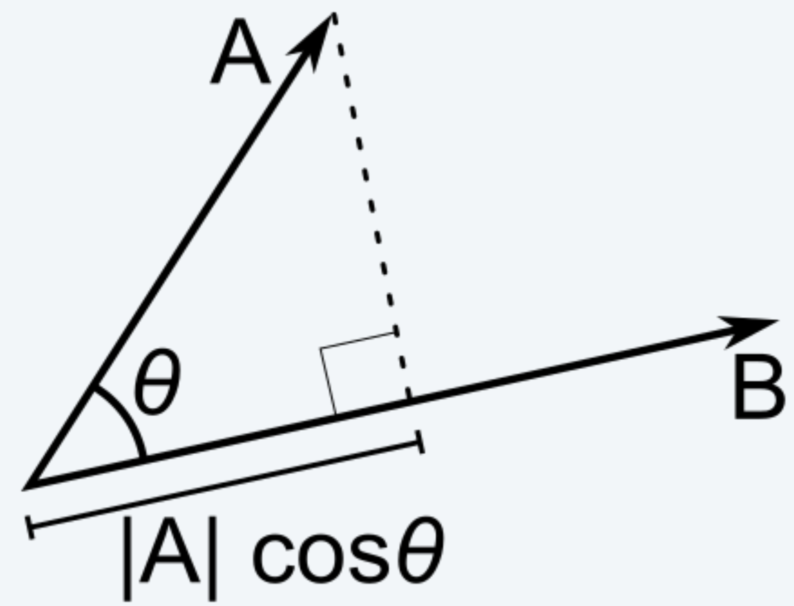
Geometric primitives

Dot product:

- $A = (a_1, a_2)$, $B = (b_1, b_2)$
- $A \bullet B = a_1 b_1 + a_2 b_2$

Geometric interpretation:

- $A \bullet B = \|A\| \|B\| \cos(\theta)$



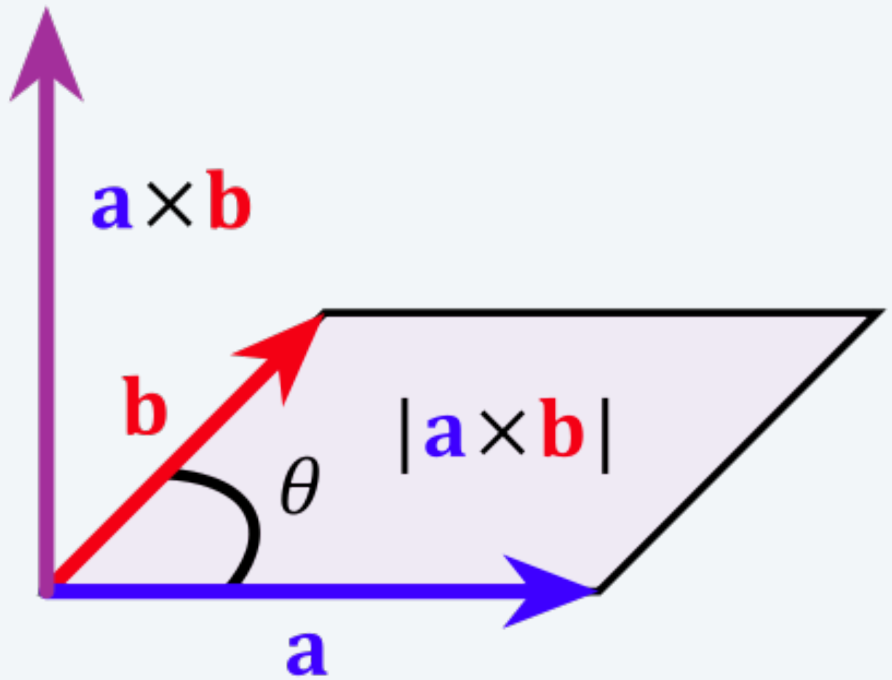
Applications:

- Finding vector length:
  - $A \bullet A = \|A\|^2$
- Computing projection:
  - $|A| \cos(\theta) = A \bullet B \,/\, \|B\|$
- Finding angles:
  - $\theta = \cos^{-1}(A \bullet B \,/\, \|A\| \|B\|)$

Geometric primitives (cont'd)

Cross product:

- $A = (a_1, a_2)$, $B = (b_1, b_2)$
- $A \times B = a_1 b_2 - a_2 b_1$



Geometric interpretation:

- $A \times B = \|A\| \, \|B\| \sin(\theta)$

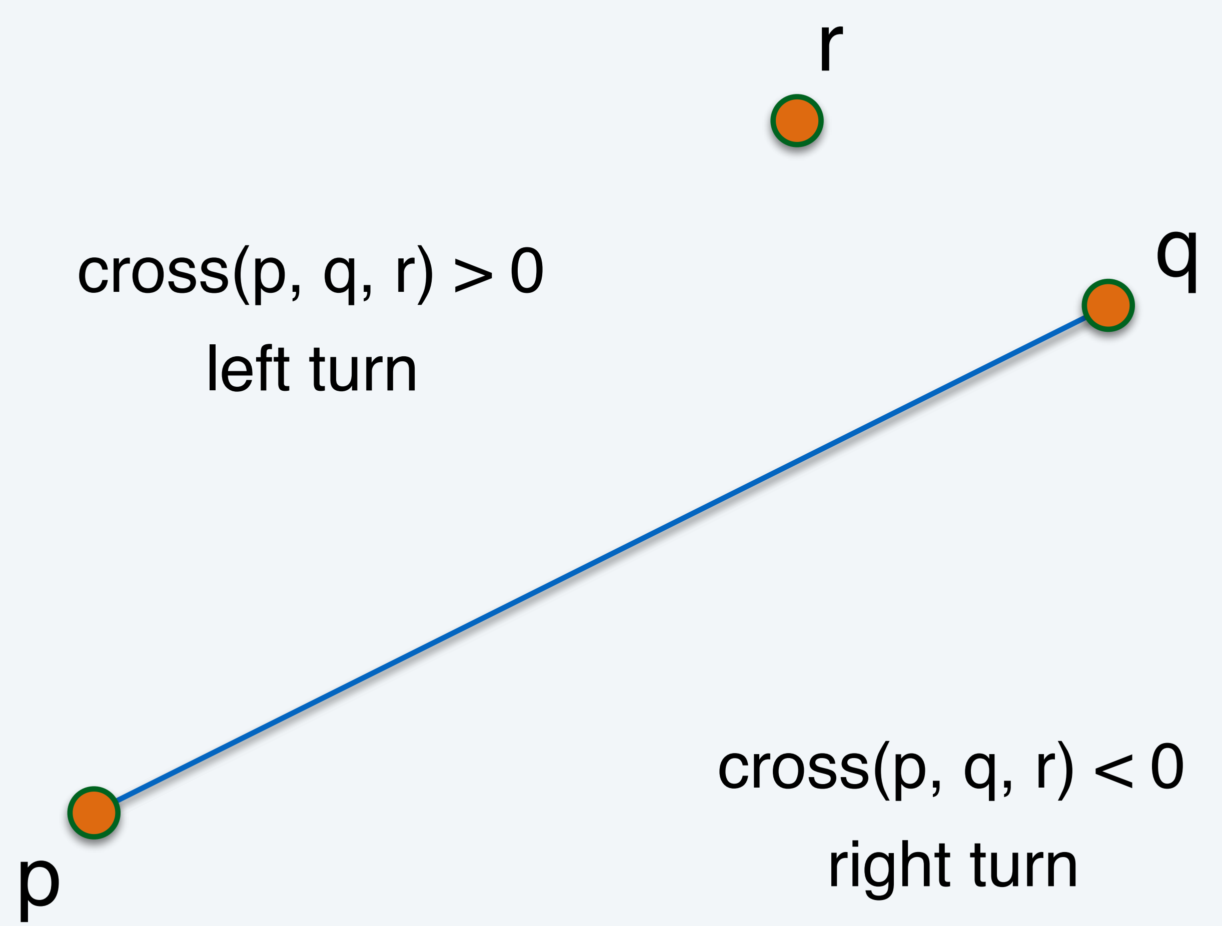
Applications:

- Computing area of a triangle:
  - $|A \times B| \;/\; 2$
- Determining the relative orientation of two vectors:
  - $A \times B = -(B \times A)$
- Distance of point r to the line through p and q:
  - $(q - p) \times (r - p) \;/\; \|q - p\|$

Orientation of three points

Define:

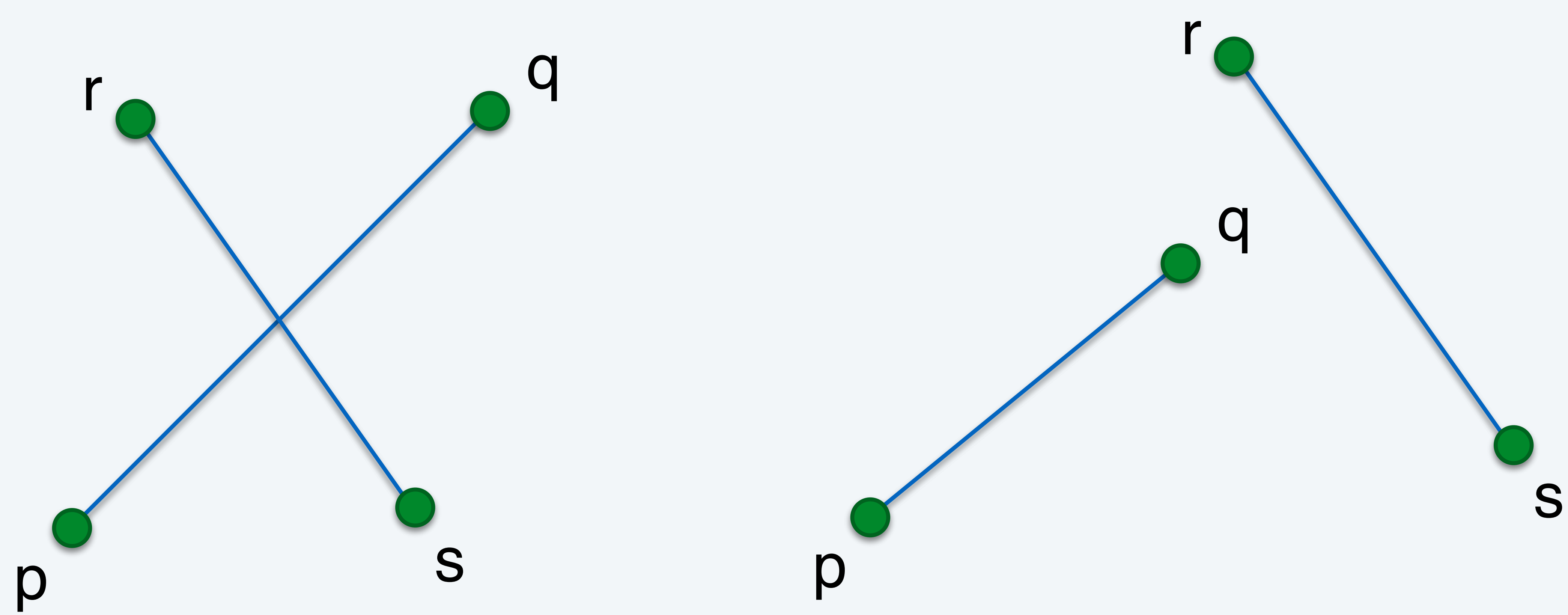
- cross(p, q, r) = (q − p) × (r − p)



Segment-segment intersection test

Proper intersection:

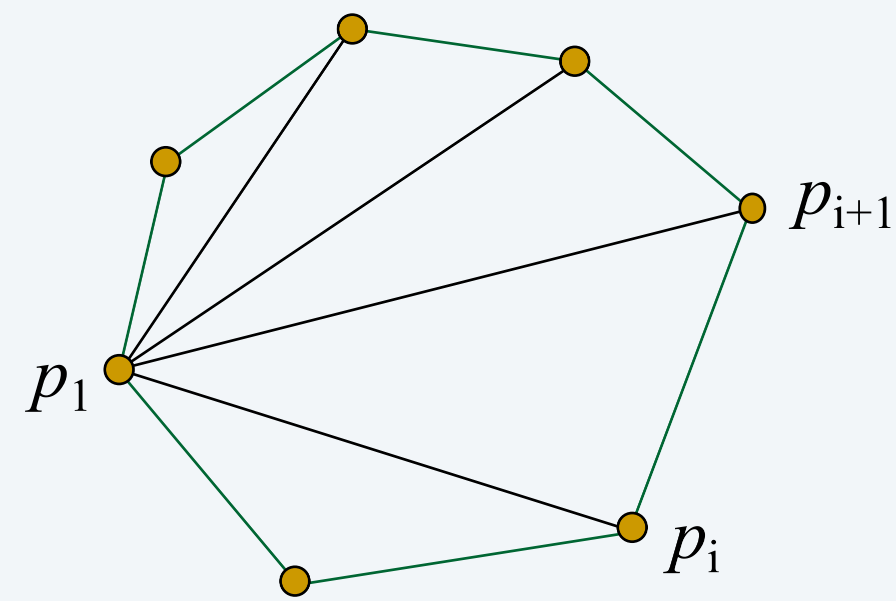
- cross(p, q, r) · cross(p, q, s) < 0
- and cross(r, s, p) · cross(r, s, q) < 0



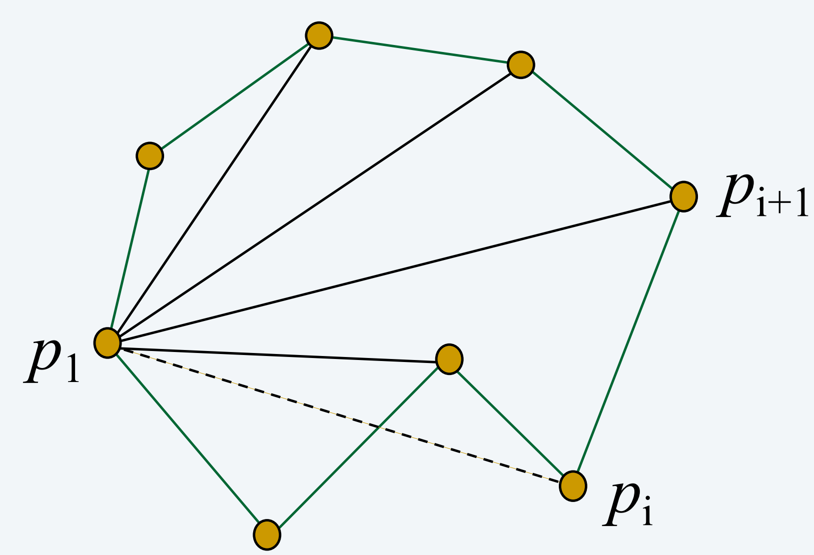
Area of a simple polygon

Problem:

- Given $p_1, p_2, \ldots, p_n$ around perimeter of a polygon P
- If P is convex, we can decompose it into triangles:
  - 2 area = $|\Sigma_{i=2..n-1} (p_i - p_1) \times (p_{i+1} - p_1)|$



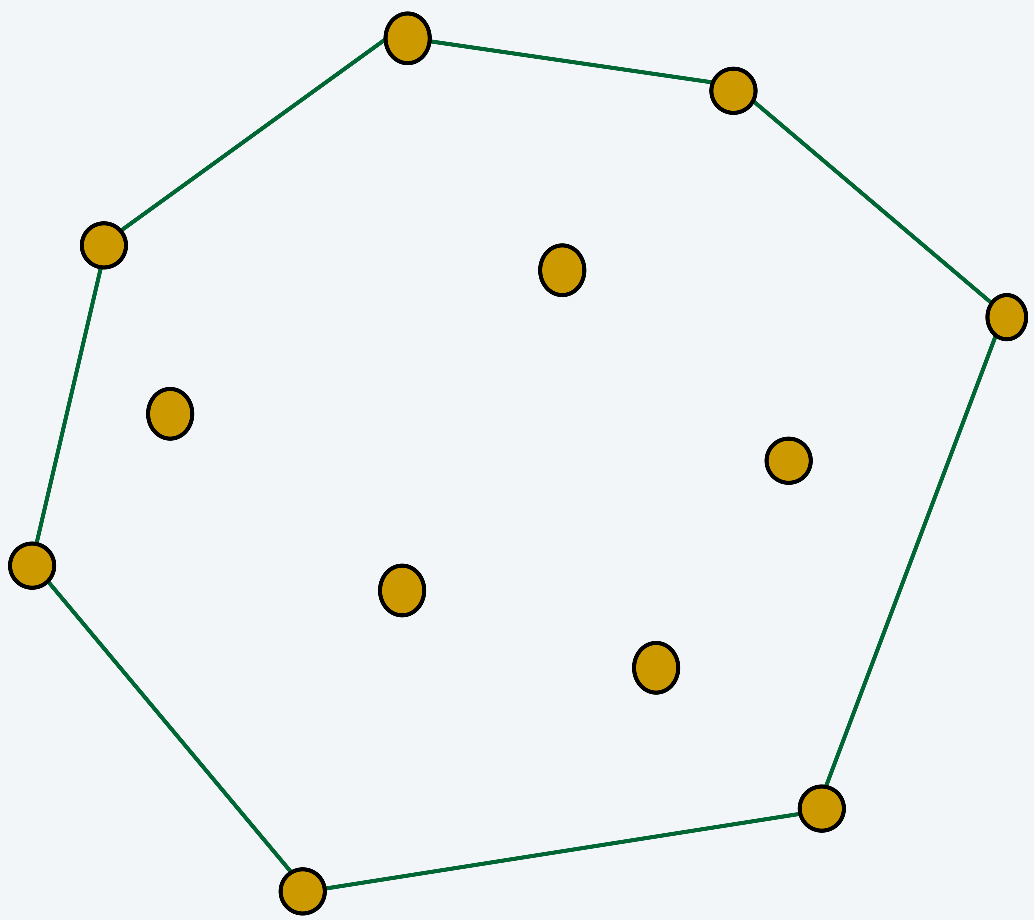
- It also works for non-convex polygons!
  - sum of "signed areas"
- Alternative formula:
  - let pi = $(x_i - y_i)$
  - 2 area = $|\Sigma_{i=1..n} (x_i - y_{i+1}) \times (x_{i+1} - y_i)|$



Convex hull problem

Problem:

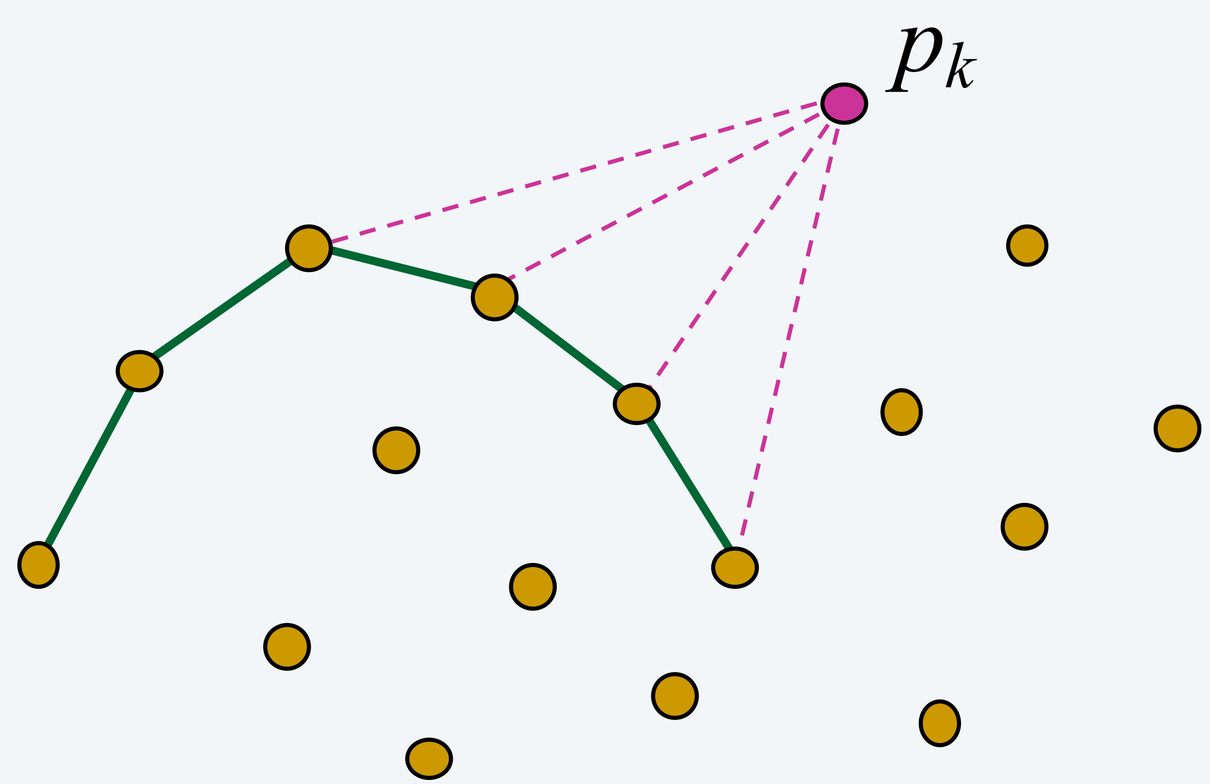
- Given n points in the plane, find the smallest convex polygon containing all the points.



Graham scan

Algorithm:

- sort points in x-coordinates: $p_1, p_2, \ldots, p_n$.
- start with an empty chain
- **for** k = 1 to n:
  - **while** the last two points of the chain and p_k make a left turn:
    - remove the last point from the chain
  - add p_k to the chain
- return the chain



Graham scan

Algorithm:

- sort points in x–coordinates: p_1, p_2,…, p_n.
- start with an empty chain
- **for** k = 1 to n:
  - **while** the last two points of the chain and p_k make a left turn:
    - remove the last point from the chain
  - add p_k to the chain
- return the chain

Analysis:

- Each point is added to chain exactly once, and is deleted at most once.
  - So the scan takes O(n) time.

- But we need to sort the points at first.
  - Overall runtime is O(n log n).

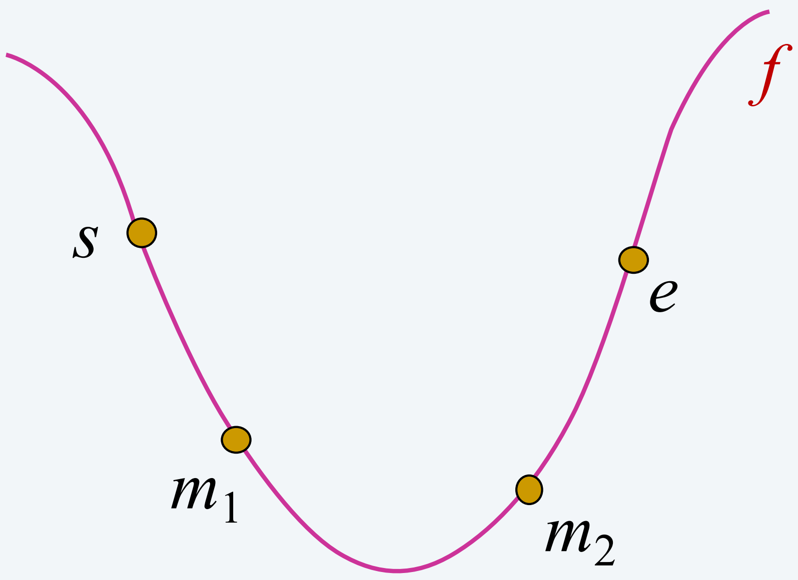
Ternary search

Problem:

- Find the minimum of a convex function f

Solution:

- start with search interval [s, e]
- **while** |e – s| is not small enough:
  - let $m_1 = s + (e - s)/3$
  - let $m_2 = e - (e - s)/3$
  - if $f(m_1) < f(m_2)$, then set e to m_2
  - otherwise set s to m_1



Team reference document

KTH Royal Institute of Technology

Omogen Heap

Simon Lindholm, Johan Sannemo, Mårten Wiman

https://github.com/kth-competitive-programming/kactl/

References

- J. Park, Introduction to Programming Contests, Stanford, Winter 2012.
- M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, Computational Geometry: Algorithms and Applications, 3rd edition, Springer, 2008.
- J. O'Rourke, Computational Geometry in C, 2nd edition, Cambridge University Press, 1998.