



Walchand College of Engineering

(Government Aided Autonomous Institute)
Vishrambag, Sangli. 416415



Computer Algorithms

String Matching & Computational Geometry

23-24

D B Kulkarni

Information Technology Department



Agenda

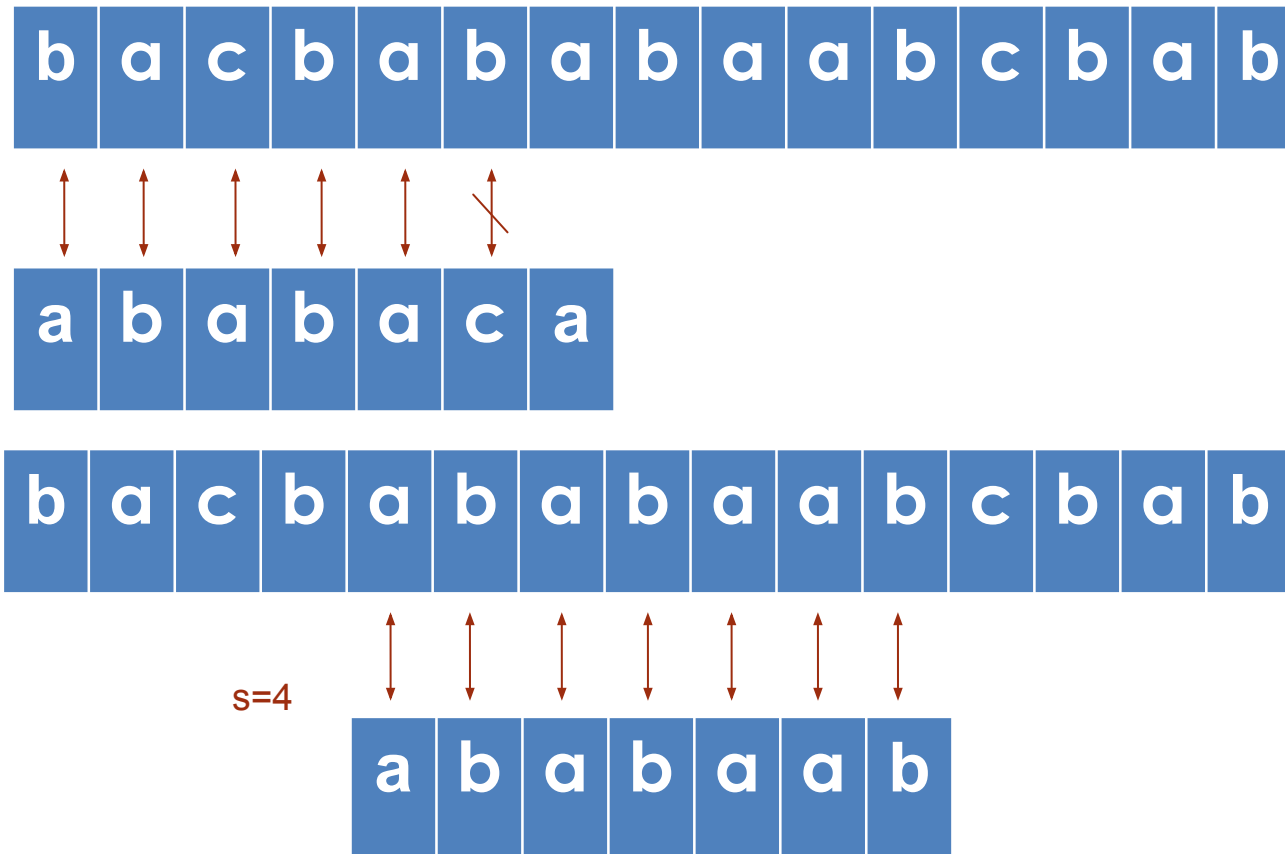
- Introduction string comparison
- Rabin Karp
-



String Matching

Given Text $T[1..n]$, Pattern $P[1..m]$ elements of T and P are drawn from Σ e.g. $\Sigma = \{0,1\}$ or $\Sigma = \{a,b,..z\}$

- Pattern P occurs with shift s in T (p occurs beginning at position $s+1$ in text T) if $0 \leq s \leq (n-m)$ and $T[s+1,s+2,...s+m]=P[1..m]$





Terminology

- Σ^* be set of all finite length strings formed using Σ
- ϵ - be zero length empty string
- Length of string x is $|x|$
- Prefix/suffix
- String x is suffix of z , is denoted by $x]z$
- String y is prefix of z , is denoted by $y[z$
- Overlapping suffix lemma
Suppose x, y, z are strings such that $x]z$ and $y]z$ then
 - If $|x| \leq |y|$ then $x]y$
 - If $|x| \geq |y|$ then $y]x$
 - If $|x| = |y|$ then $x = y$



Rabin Karp

Challenges

- Decide prime number

Applications

- Plagiarism

- Multiple pattern matching

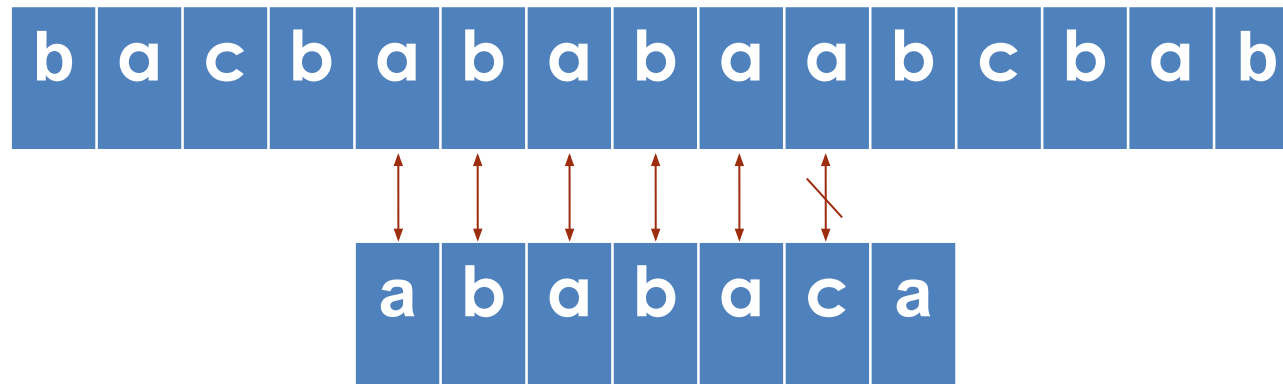
Modified

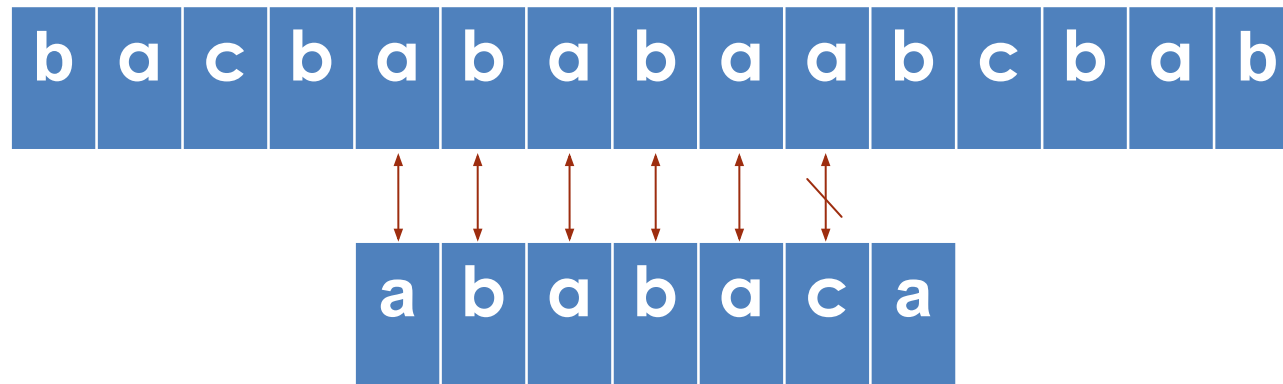
- Extend Rabin Karp for 2D pattern matching

 - $n \times n$ text with $m \times m$ pattern



- How will this technique be applicable for text characters?
- Map characters to digit
- E.g. if $\Sigma=\{a,b,c,d,e\}$ is mapped on to $\{1,2,3,4,5\}$ then
 $T=abecda = \{125341\}$ and $P=cda = \{341\}$
- Use Hash function 3^n
- $P=cda \rightarrow \text{Hash} \rightarrow 341 \sim 3 \times 3^0 + 4 \times 3^1 + 1 \times 3^2 = 24$
- Take abe part of $T=\{125\} \rightarrow \text{Hash} \rightarrow 52$ which is not equal to 24
- Take the next digit 3- $\{253\} \rightarrow \text{Hash} = (52-1)/3 + 3 \times 3^2 = 44$
- Take the next digit 4- $\{534\} \rightarrow \text{Hash} = (44-2)/3 + 4 \times 3^2 = 50$
- Take the next digit 1- $\{341\} \rightarrow \text{Hash} = (50-5)/3 + 1 \times 3^2 = 24$ Match after individual comparison





Find longest prefix which is proper suffix of P



Knuth Morris Pratt (KMP) Algorithm

a	b	a	b	a	c	a
---	---	---	---	---	---	---

Find longest prefix which is proper suffix of P

Use auxiliary function π , which is precomputed from pattern

$\pi =$

0	0	1	2	3	0	1
---	---	---	---	---	---	---



KMP

b	a	c	b	a	b	a	b	c	a	b	c	b	a	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

a	b	a	b	a	c	a
---	---	---	---	---	---	---

0	0	1	2	3	0	1
---	---	---	---	---	---	---

$k=4; \pi(3)=2$

Shift pattern by 2

Start matching

b	a	c	b	a	b	a	b	c	a	b	c	b	a	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

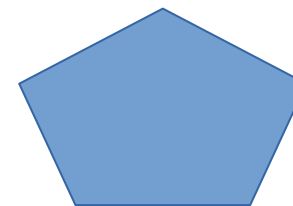
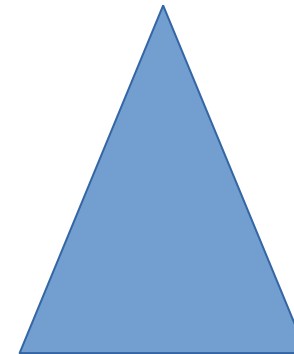
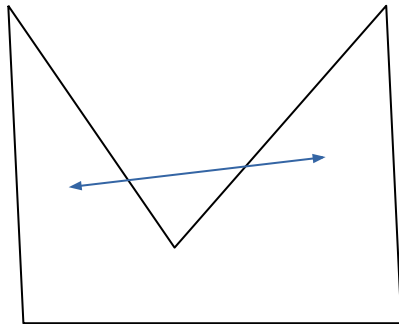
a	b	a	b	a	c	a
---	---	---	---	---	---	---

0	0	1	2	3	0	1
---	---	---	---	---	---	---



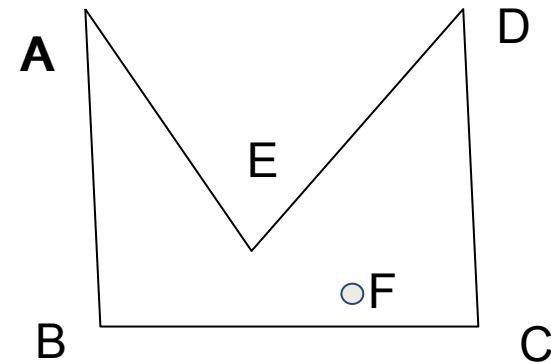
Convex Polygon

- Definition: If we join any two point within polygon, all the points lying on the line are within the polygon.





- Check if point F is within polygon ABCDE



Problem statement: How will you form a convex polygon out of n given vertices?

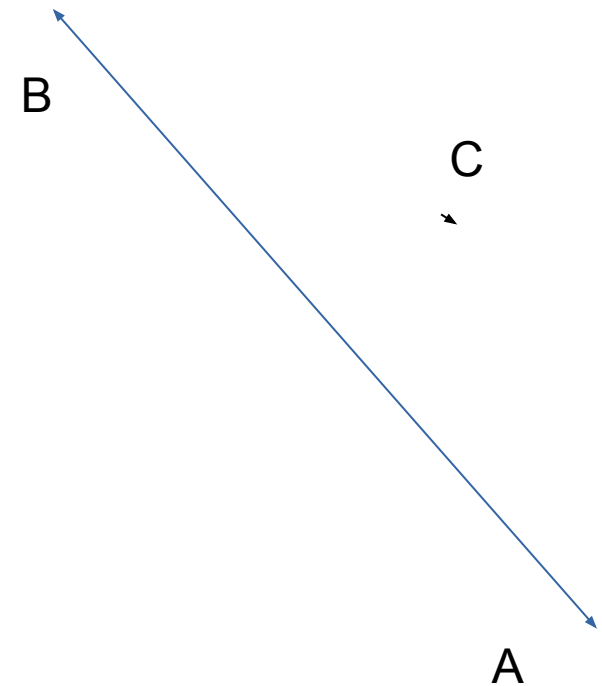
- Vertices forming convex polygon, k where ($k \ll n$)
- Order in which these k points be connected



Convex Polygon

- Given set of points (n - with x and y coordinates), find whether the polygon defined by n points is convex?
 - Order of points is important!
 - Find if the a given point is within the polygon....?

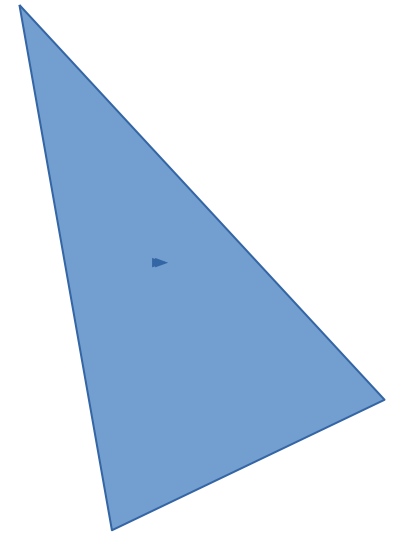
Given a line (AB) and a point C, decide to which side of the the line, the point lies?





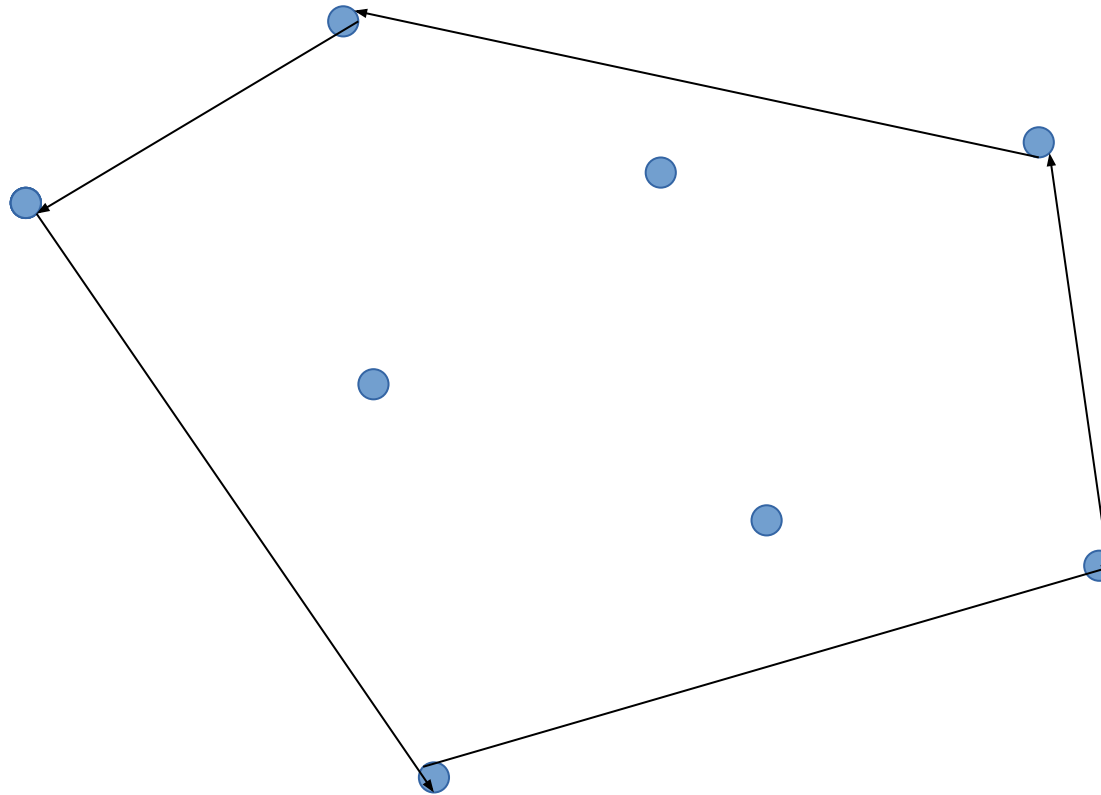
Solution: Triangles that can be formed out of given n points
 $O(n^3)$

- Check for every point, if it lies within any one of these triangles
- Assume that checking if point lies within triangle takes one unit
- There are n points, so total time $O(n^4)$





Divide and Conquer (D&C)

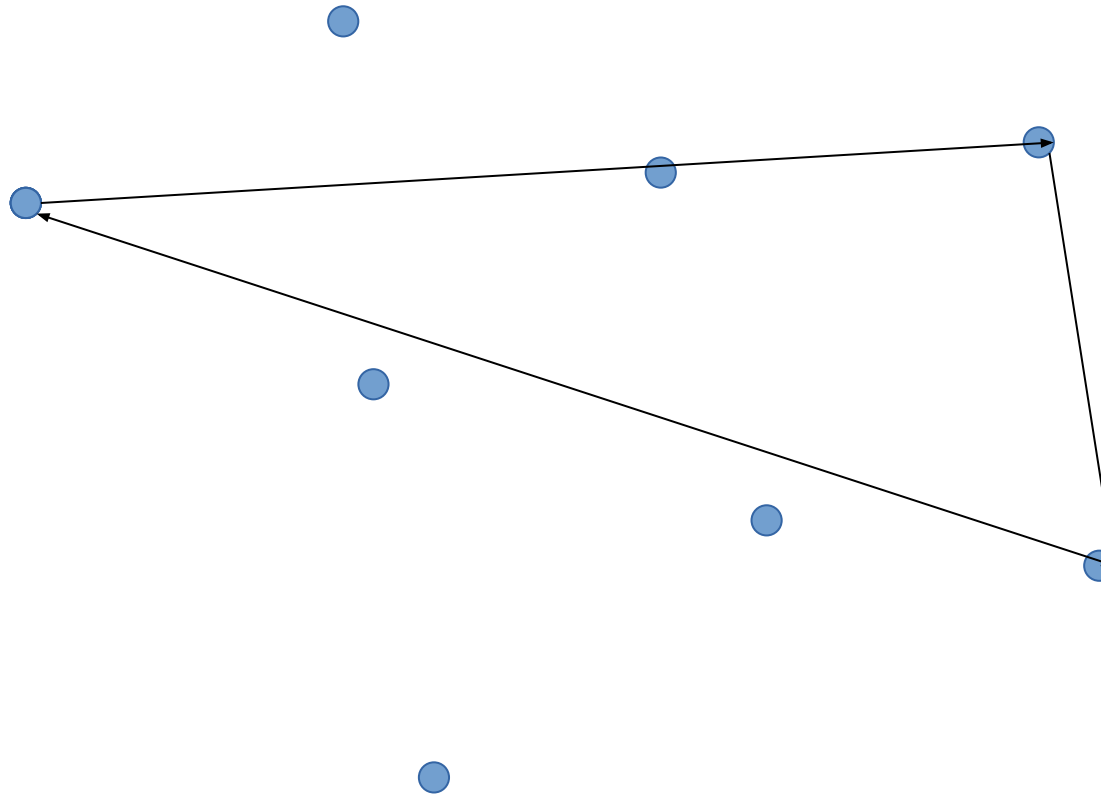




- Select extreme points (left, right)
- Line joining these points divides the space in lower/upper hull
- Upper Hull
 - Select triangle formed by these two points and third point where the area is max
 - Remove all points in this triangle
 - Consider two smaller hulls above the sides of triangle
 - Repeat till only one point left
- Repeat the process for lower Hull



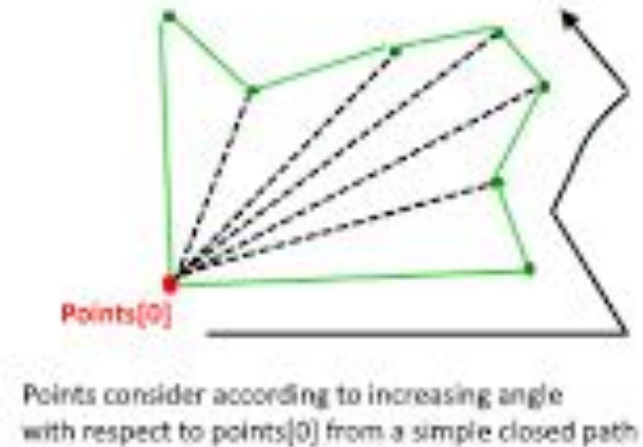
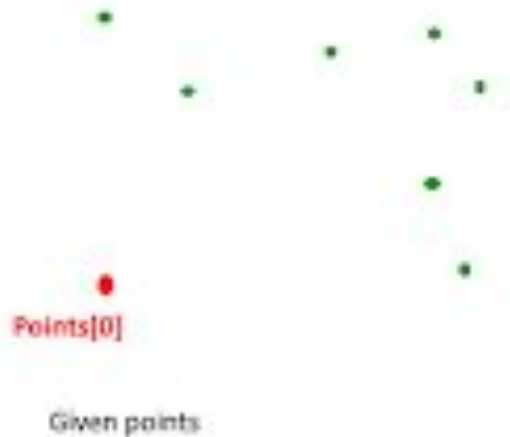
Upper hull



Complexity: $n \log n$



Other methods: Graham scan



- Note lowest and highest point
- Start from lowest point
- At every point search for point which makes lowest polar angle with +ve X axis
- Complexity $n \cdot h$ (h - points defining convex polygon)



Other methods: Jarvis march

- Start from lowest point
- Sort all points based on their increasing polar angles with +ve X axis
- Consider 3 successive points p_1, p_2, p_3
- If p_3 is to the left of P_1p_2 include p_2 in list of points defining CH
- Consider next point in order p_4
($p_2p_3p_4$)-> $p_1p_2p_3$
- Recurse
- Complexity $n \log n$

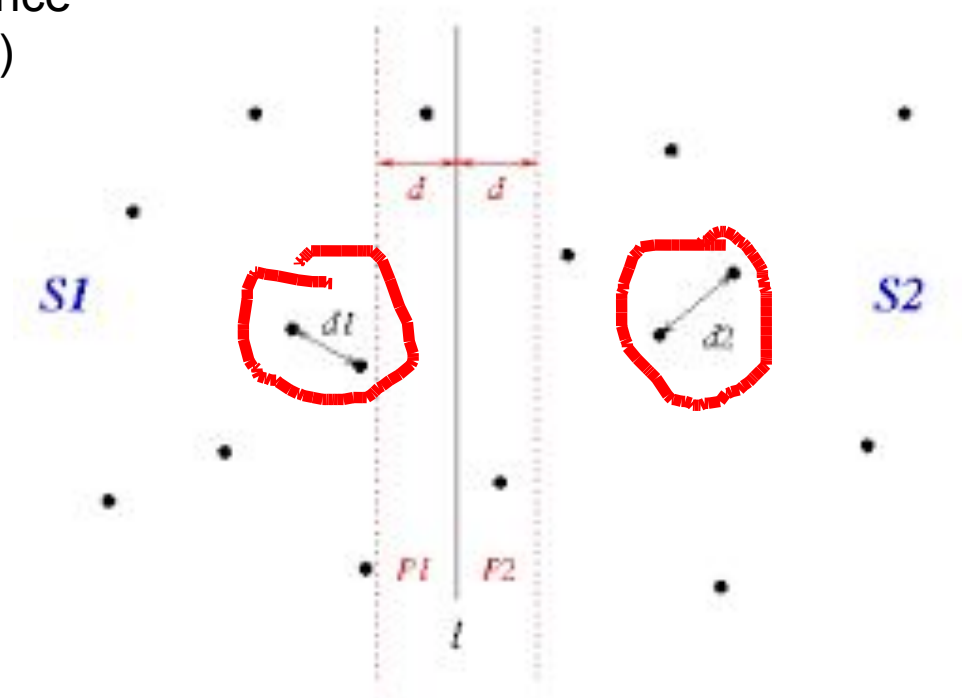


Closest pair of points algorithm

Consider all possible pair of points

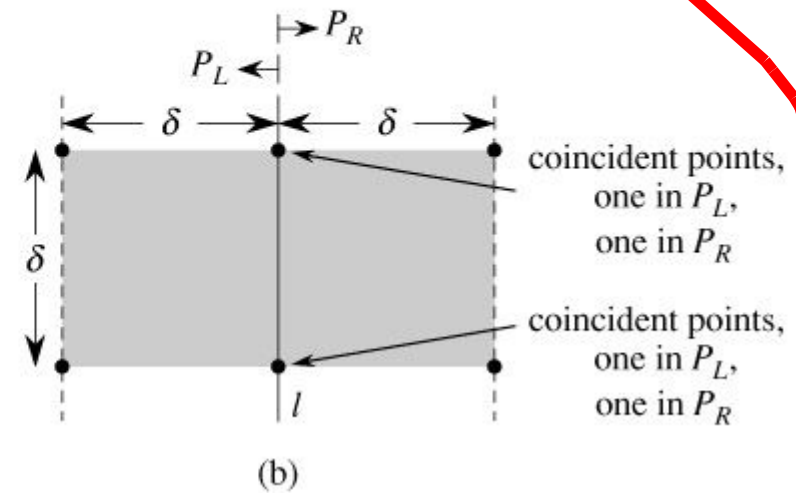
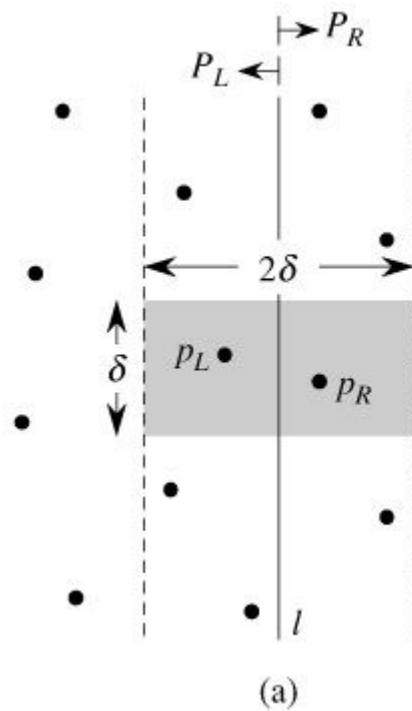
Compute the distance

Complexity: $O(n^2)$





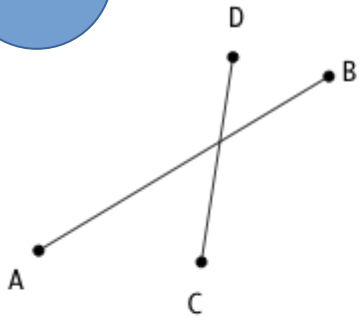
Closest pair of points algorithm



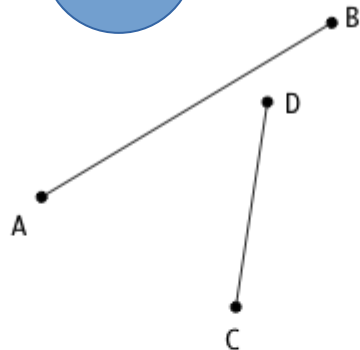


Pair of intersecting segments

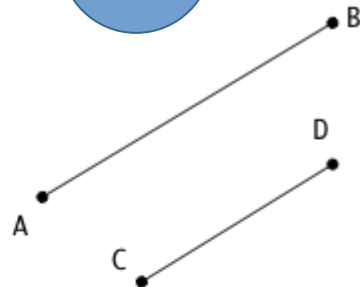
1



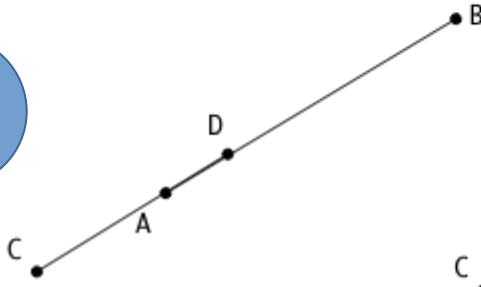
2



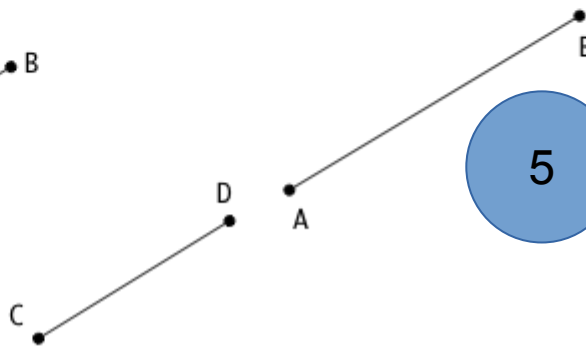
3



4



5

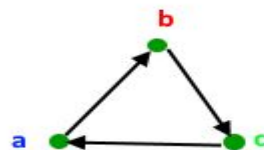


- Intersecting
- Non intersecting
- Parallel
- Colinear (overlapping)
- Colinear

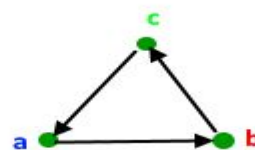


Orientation

- Orientation of an ordered triplet of points in the plane can be
 - Counterclockwise
 - Clockwise
 - colinear
- The following diagram shows different possible orientations of (a, b, c)



Clockwise



Counterclockwise



Collinear



Intersection

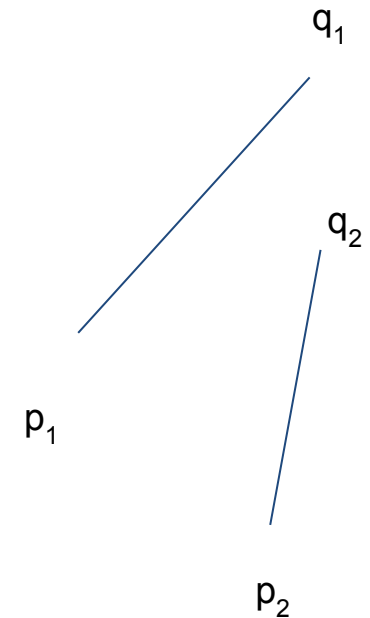
- Two segments (p_1, q_1) and (p_2, q_2) intersect if and only if one of the following two conditions is verified

General Case:

- (p_1, q_1, p_2) and (p_1, q_1, q_2) have different orientations
and
- (p_2, q_2, p_1) and (p_2, q_2, q_1) have different orientations.

Special case:—

- (p_1, q_1, p_2) , (p_1, q_1, q_2) , (p_2, q_2, p_1) , and (p_2, q_2, q_1) are all collinear
and
- the x-projections of (p_1, q_1) and (p_2, q_2) intersect
- the y-projections of (p_1, q_1) and (p_2, q_2) intersect





Intersection of line-segments

- Bruteforce method
 - For every segment
 - Check if it intersects with remaining others
- Complexity
 - $n(n-1)/2$
- Sweepline
 - Scans from L-R
 - Generates event
 - Insert
 - Delete
 - Intersect



Whether pair of segments intersect?

