Walchand College of Engineering

(Government Aided Autonomous Institute)

Vishrambag, Sangli. 416415

Computer Algorithms

String Matching & Computational Geometry

24-25

D B Kulkarni

Information Technology Department

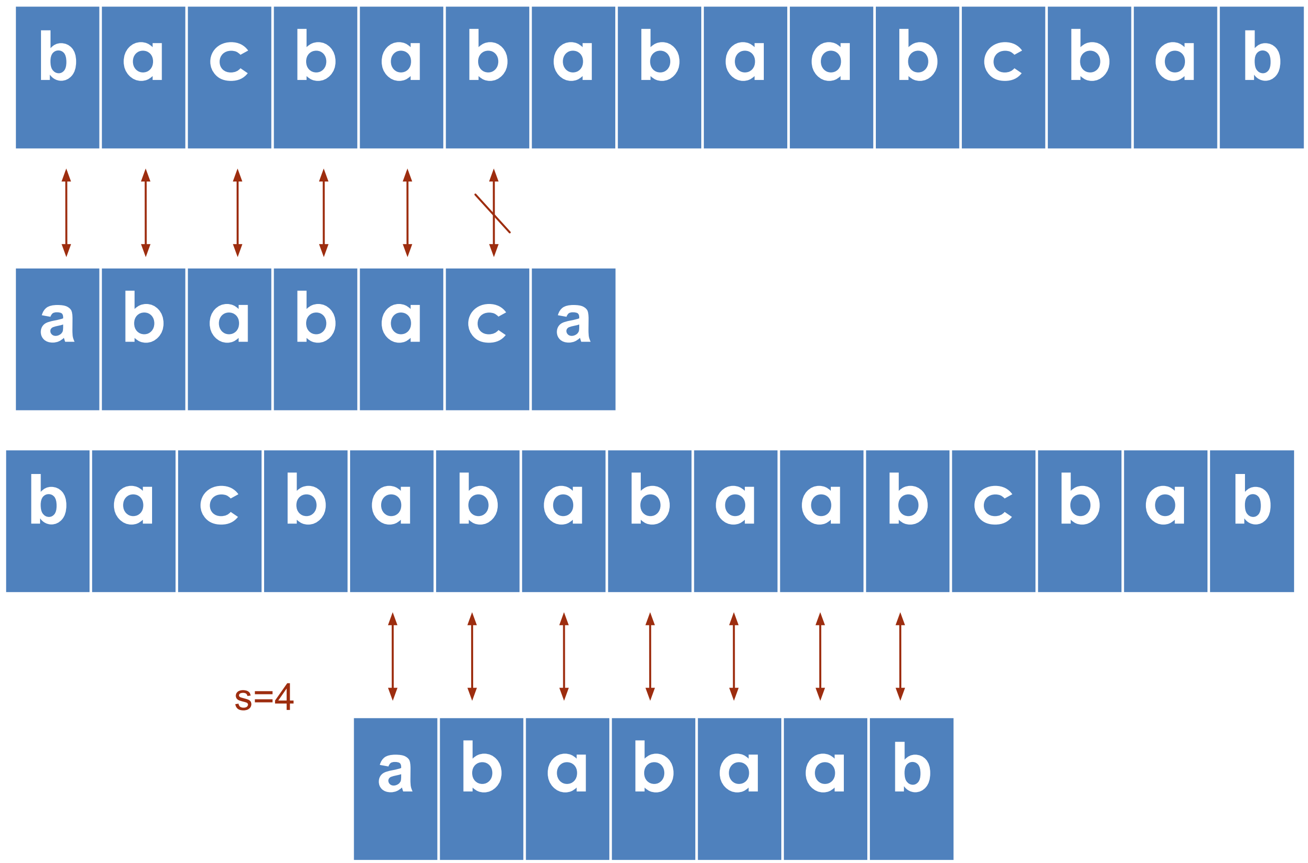
Agenda

- Introduction string comparison
- Rabin Karp
-

String Matching

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

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



Terminology

- $\sum^*$ be set of all finite length strings formed using $\sum$
- ε- be zero length empty string
- Length of string x is |x|

Preceding (Prefix) and Following (suffix)

- String w is called a prefix of string s if s = wy for some string y in Σ*. This is denoted w ⊏ s
- String w is a suffix of s, denoted w ⊐ s, if s = yw for some y in Σ*.

- Example, ax ⊏ axiometric and metric ⊐ axiometric.

- Overlapping suffix lemma

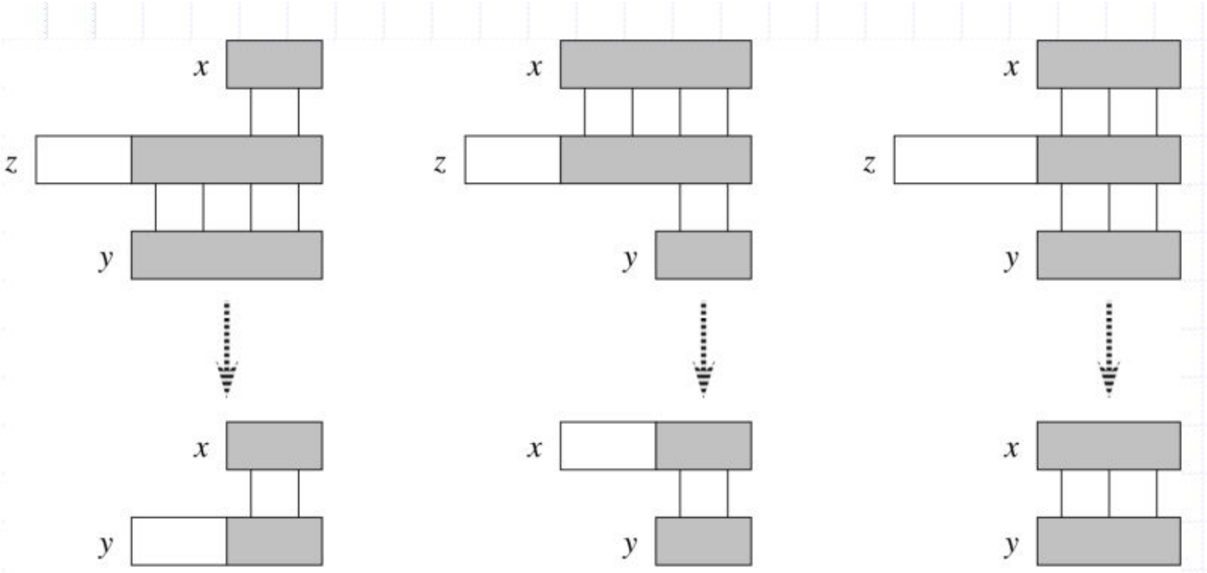
  Suppose x,y,z are strings such that

  x]zand y]z then

  If |x|≤|y| then x]y

  If |x|≥|y| then y]x

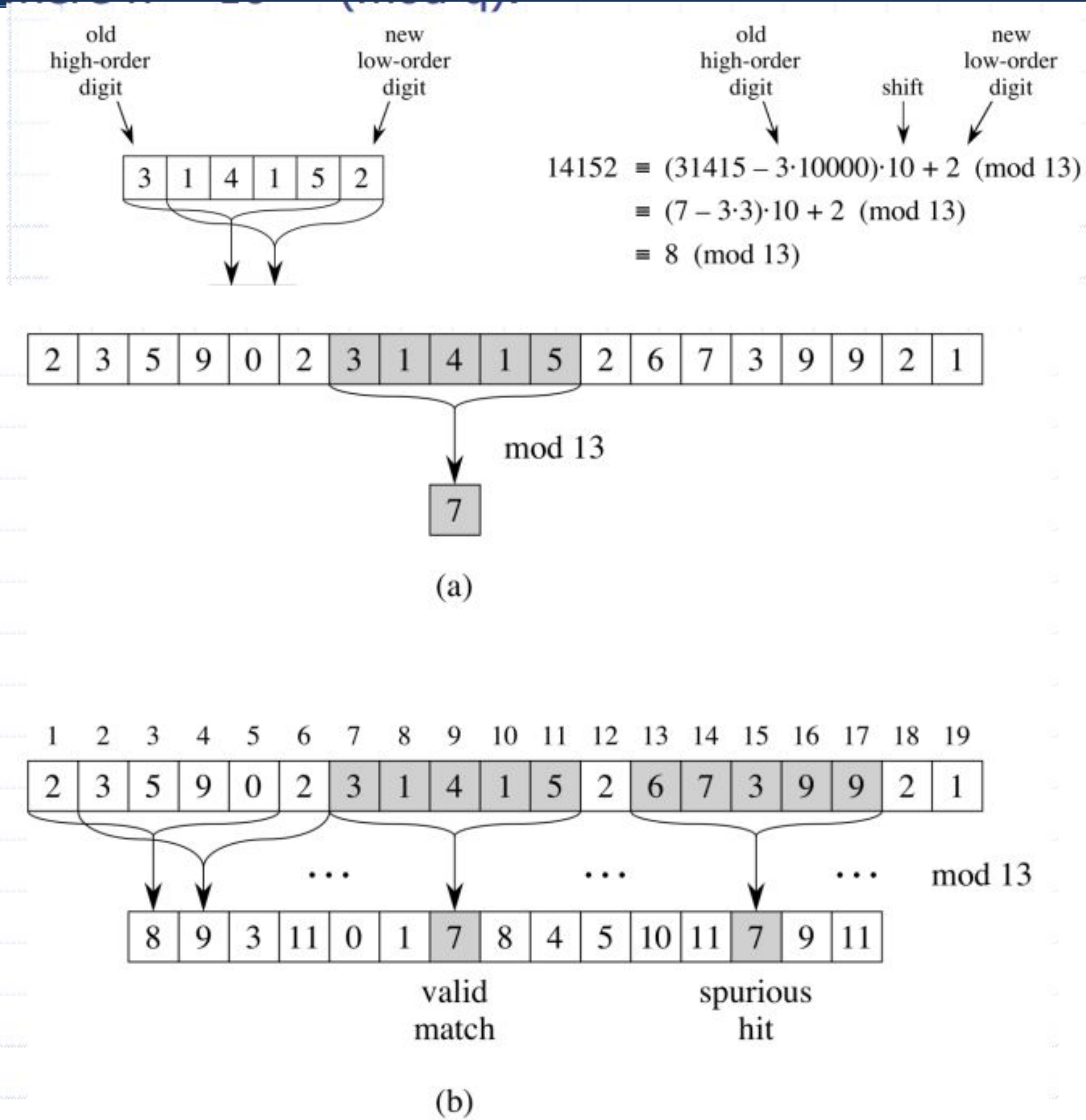
  If |x|=|y| then x=y



Rabin Karp

- Rabin Karp
  - Challenges
    - Decide prime number
  - Applications
    - Plagiarism
    - Multiple pattern matching
  - Modified
    - Extend Rabin Karp for 2D pattern matching
      - n x n text with m x m pattern

Use mod function

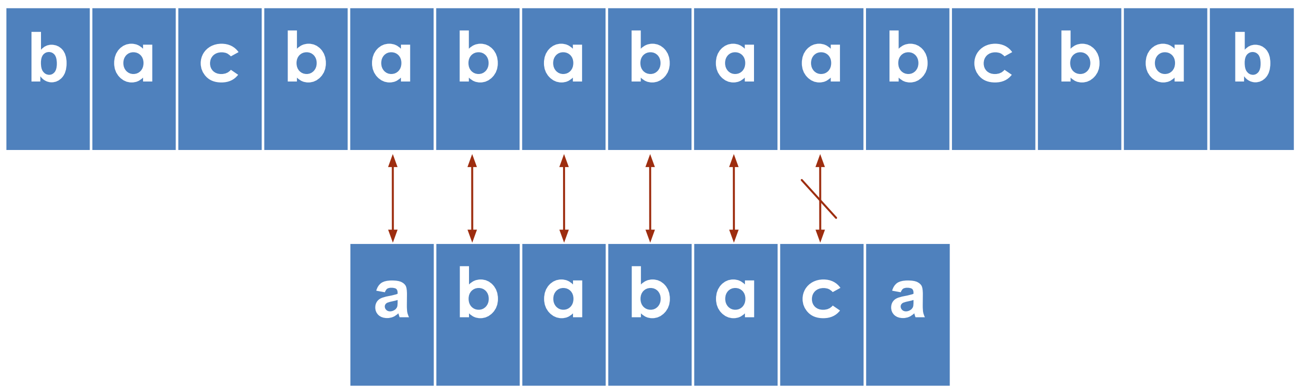


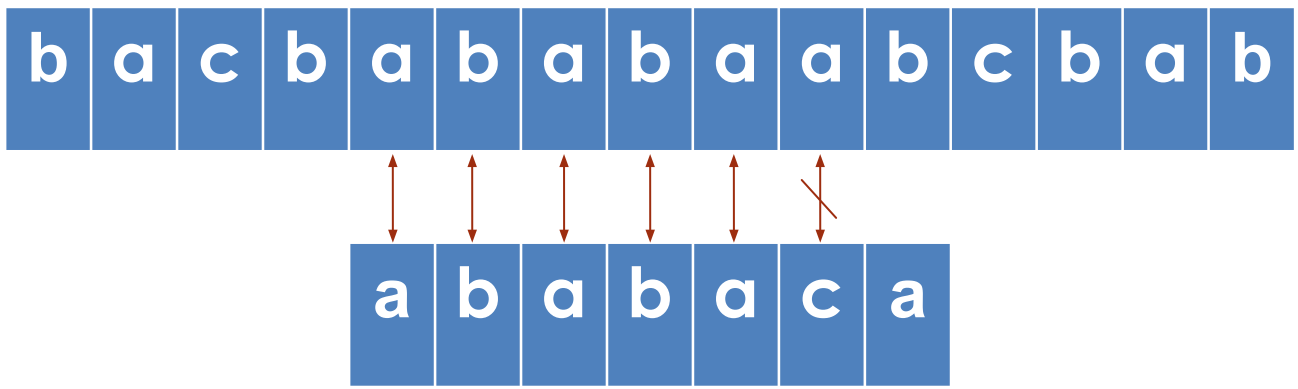
Rabin-Karp Algorithm

- How will this technique be applicable for text characters?

Using Hashing technique

- Map characters to digit
- E.g. if ∑={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->Hash>341~$3x3^0+4x3^1+1x3^2=24$
- Take abe part of T={125}->Hash>52 which is not equal to 24
- Take the next digit 3-{253}->Hash= $(52-1)/3)+3x3^2=44$
- Take the next digit 4-{534}->Hash= $(44-2)/3)+4x3^2=50$
- Take the next digit 1-{341}->Hash= $(50-5)/3)+1x3^2=24$ Match after individual comparison


b a c b a b a b a a b c b a b
a b a b a c a



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 ∏, which is precomputed from pattern

∏=

0	0	1	2	3	0	1

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; ∏(3)=2

Shift pattern by 2

Start matching

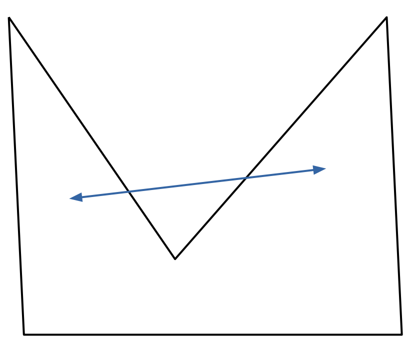
b	a	c	b	a	b	a	b	c	a	b	c	b	a	b

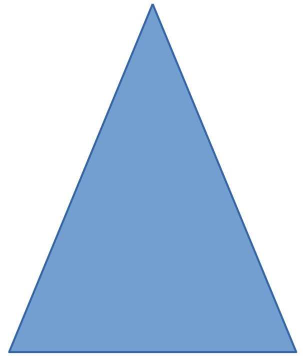
2

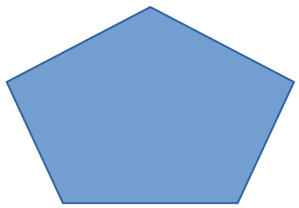
a	b	a	b	a	c	a
0	0	1	2	3	0	1

Convex Polygon

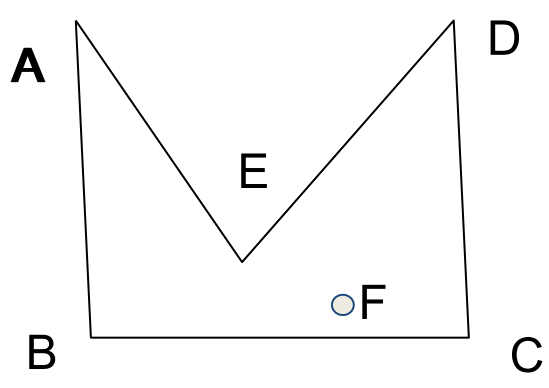
- Definition: If we join any two point within polygon, all the points lying on the line are within the polygon.







- Check if point F is within polygon ABCDE

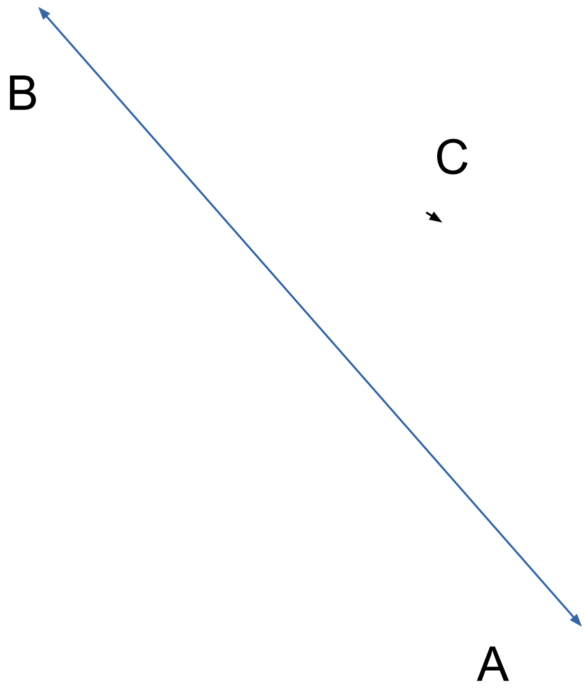


Problem statement: How will you form a convex polygon out of n given vertices?

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

Convex Polygon

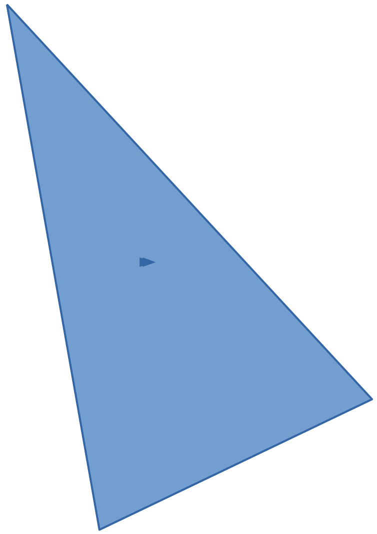
- Given set of points (n- with x and y coordinates), find whether the polygon defined by n points is convex?
  - Order of points is important!
  - Find if the a given point is within the polygon....?



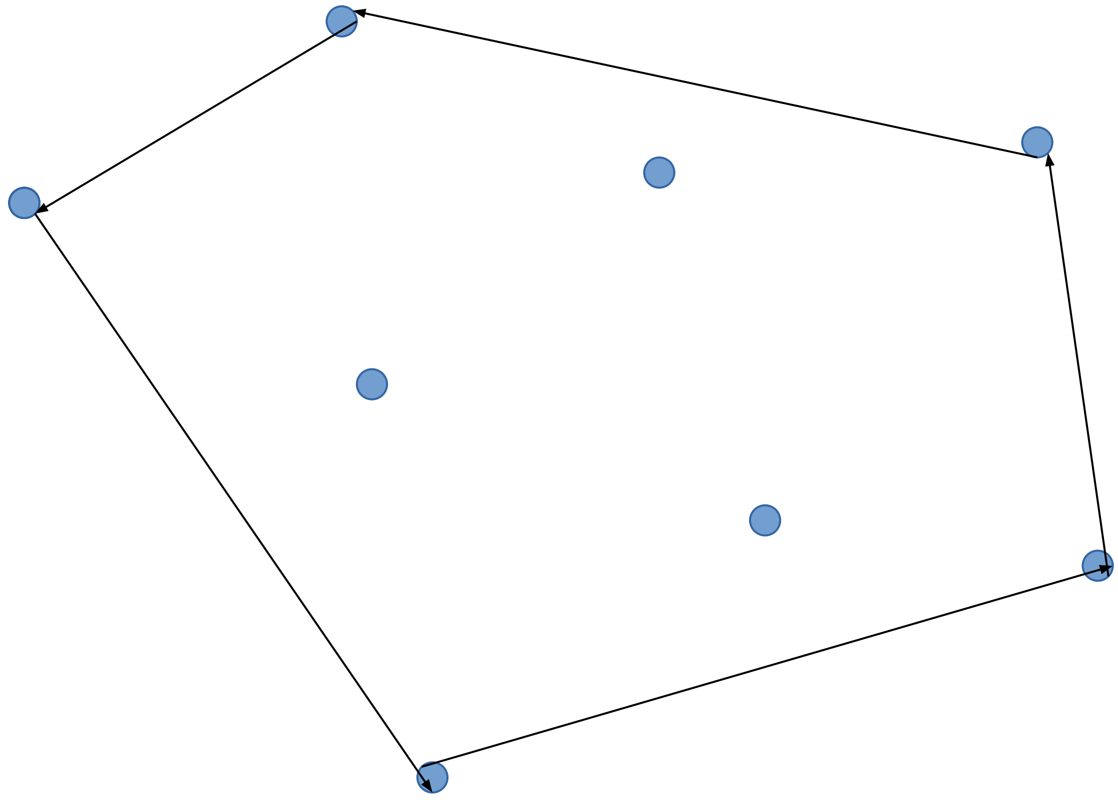
Given a line (AB) and a point C, decide to which side of the the line, the point lies?

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)



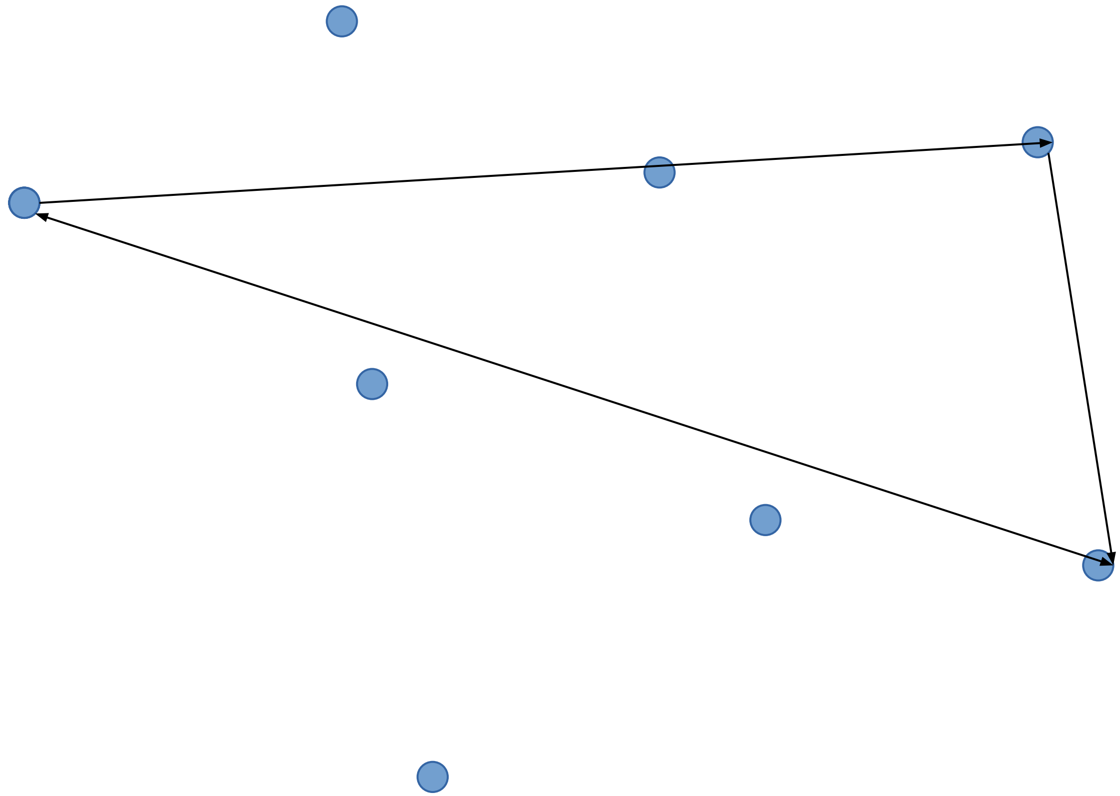
Divide and Conquer (D&C)



Procedure

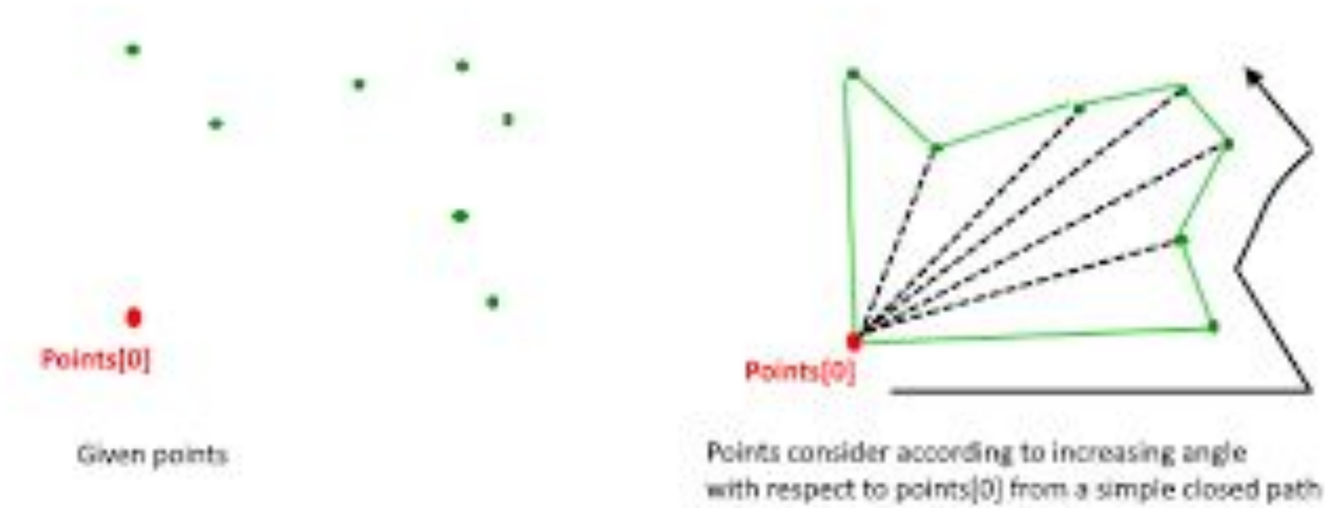
- Select extreme points (left, right)
- Line joining these points divides the space in lower/upper hull
- Upper Hull
  - Select triangle formed by these two points and third point where the area is max
  - Remove all points in this triangle
  - Consider two smaller hulls above the sides of triangle
  - 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 angli with +ve X axis

- ➢ Complexity n*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 axix
- Consider 3 successive points p1,p2,p3
- If p3 is to the left of P1p2 include p2 in list of points defining CH
- Consider next point in order p4 (p2p3p4)->p1p2p3
- Recurse

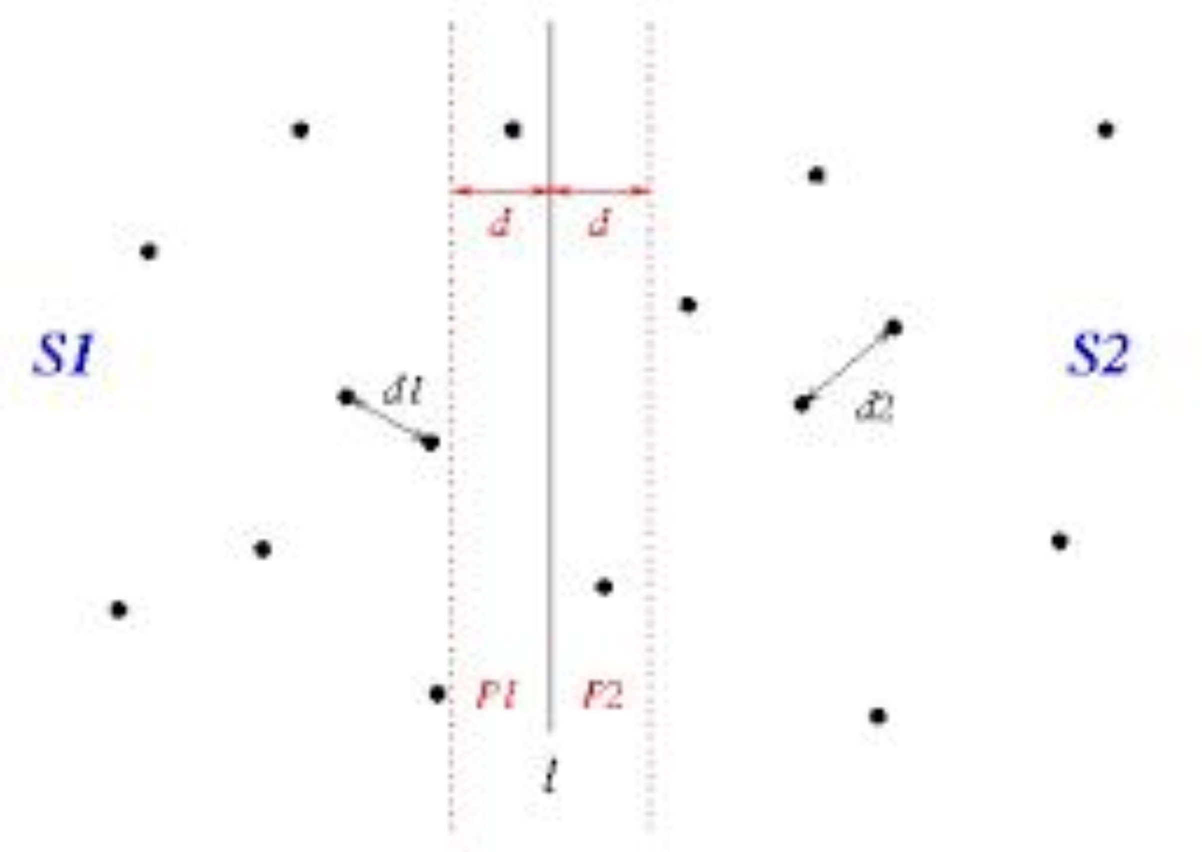
- Complexity n log n

Closest pair of points algorithm

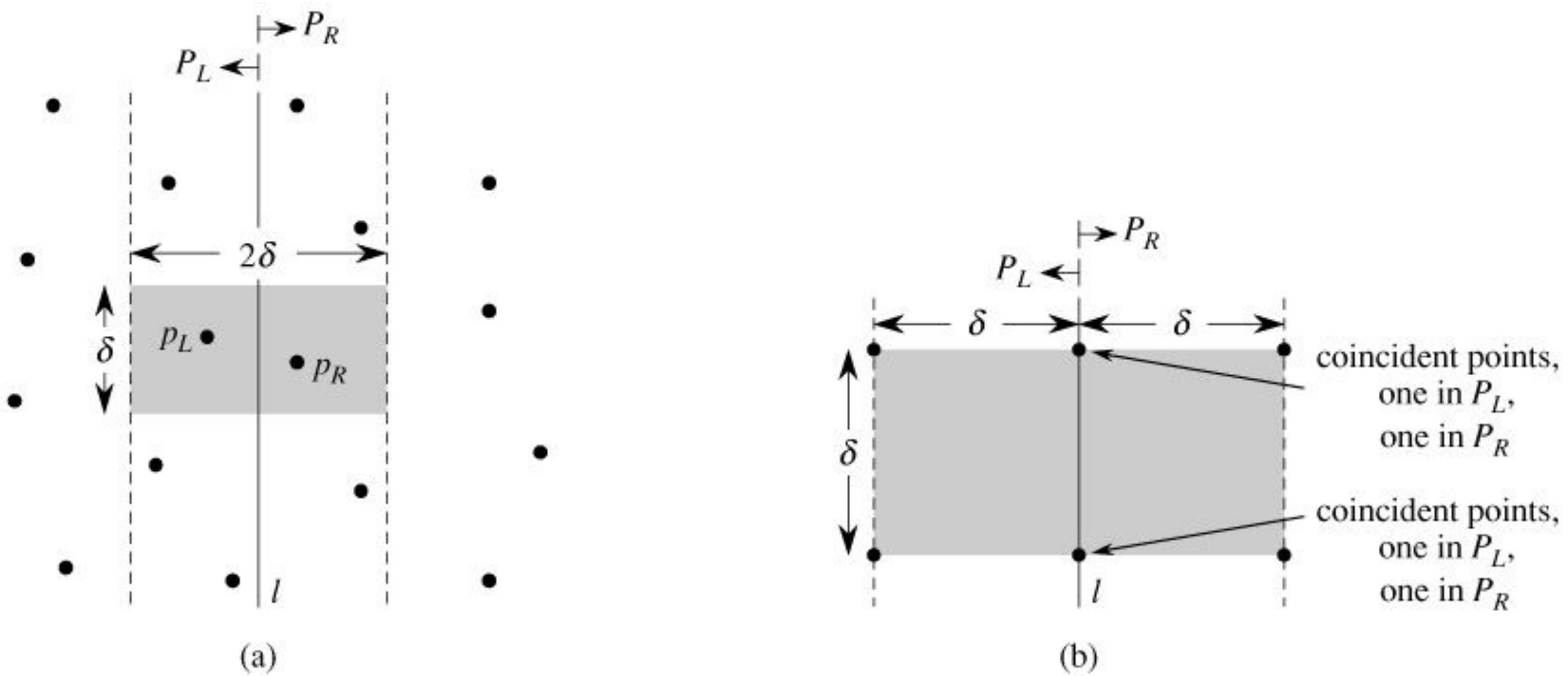
- Consider all possible pair of points
  - Take a point, calculate its distance from every other point
  - Do it for all points
- Compute the minimum distance
- Complexity: $O(n^2)$

Closest pair of points - Divide and Conquer

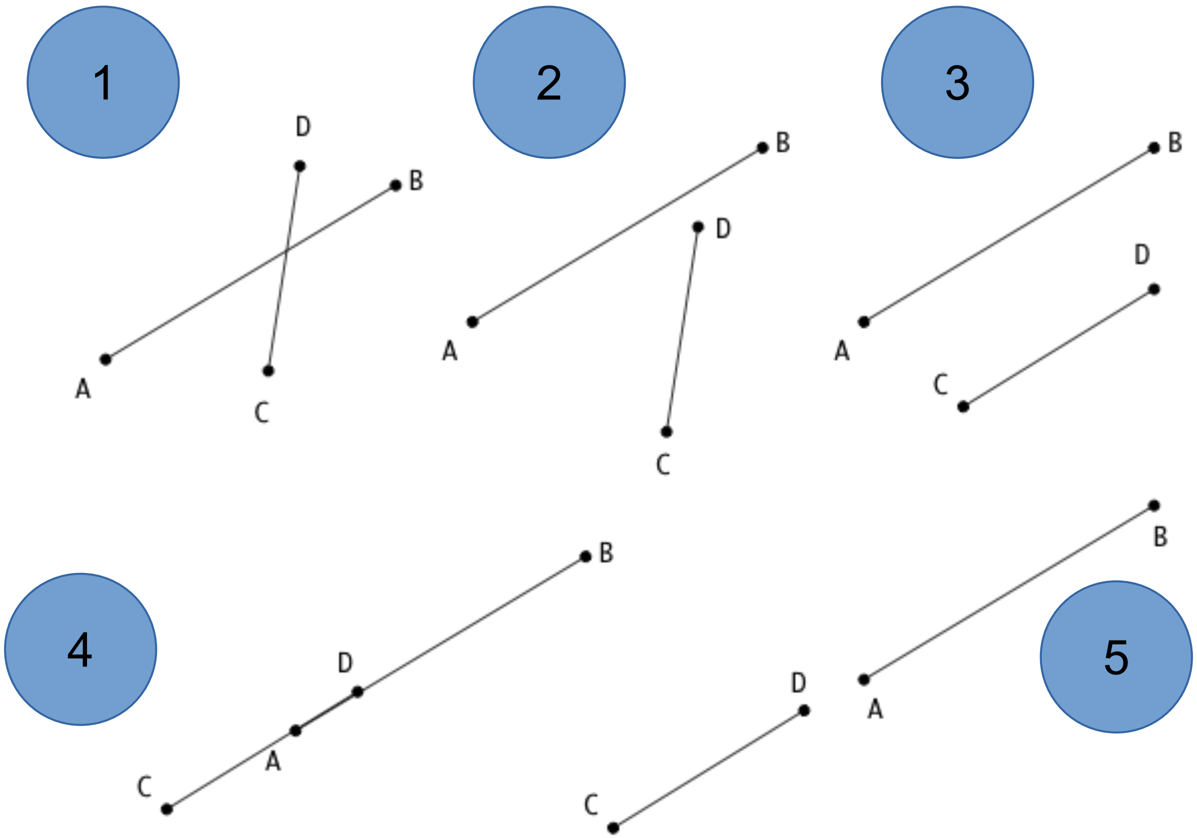
- ➢ Divide the plane vertically into Left (S1) and Right (S2) region
- ➢ Find the closest pair in S_1 and S_2 region
- ➢ Let δ_1 be distance of closest pair in S_1 and δ_2 be distance of closest pair in S_2
- ➢ Let $\delta = \min(\delta_1, \delta_2)$



Closest pair of points algorithm



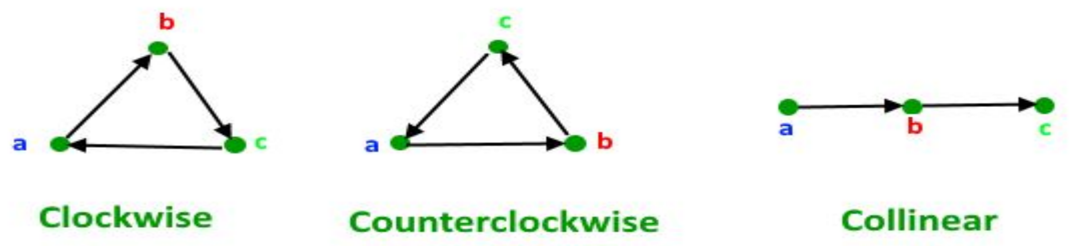
Pair of intersecting segments



- Intersecting (1)
- Non intersecting(2)
- Parallel (3)
- Colinear(overlap, 4)
- Colinear (5)

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)



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

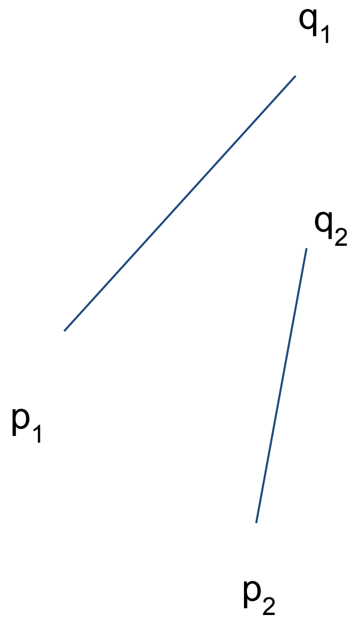
- $(p_2$, q2, $p_1)$ and (p2, q2, q1) have different orientations.

Special case:–

- (p1, q1, p2), (p1, q1, q2), (p2, q2, p1), and (p2, q2, q1) are all collinear

  and

- the x-projections of (p1, q1) and (p2, q2) intersect
- the y-projections of (p1, q1) and (p2, q2) intersect



Intersection of line-segments

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

Whether pair of segments intersect?