Line Segment Intersection

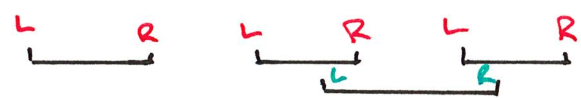
Brute force Solution

- for each pair of line segments check if that pair intersect

$O(n^2)$

1D Line segment intersection

Given a collection of n intervals determine if any overlap



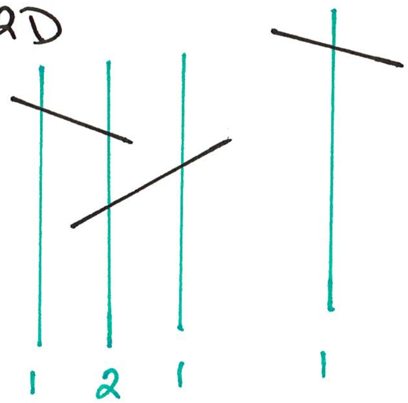
1. label each endpoint as L or R
2. Sort the endpoints
3. The intervals are disjoint iff the labels of the sorted sequence alternate

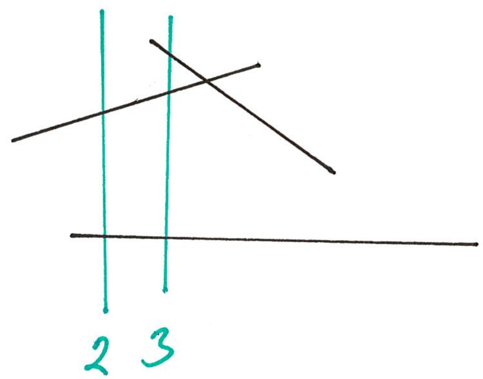
$\Theta(n \log n)$

2D Line Segment Intersection

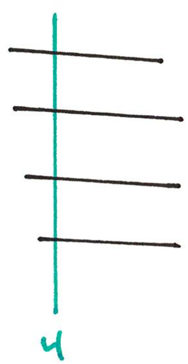
- generalize the 1D solution

- 2D

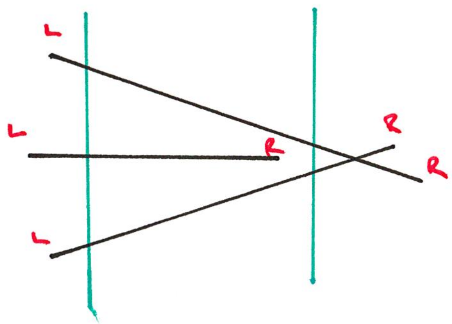


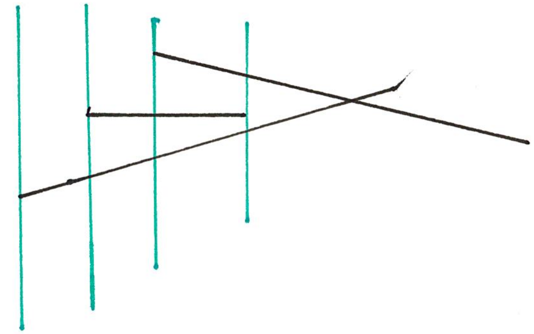


check if the lines intersecting the sweep line cross



Two lines can intersect if they become adjacent in the sweep data structure. (sorted by y-coordinate)





- sweep a vertical line across the plane from left to right
- maintain a sweep data structure to track the line segments currently intersecting the sweep line
  - sorted by y-coordinate
- sweep event point are the segment endpoints
  - a L endpoint insert the segment
  - a R endpoint removes the segment

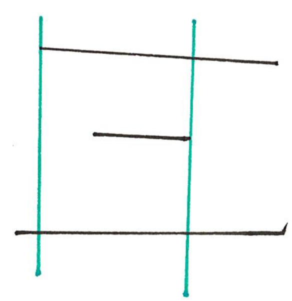
- Two line segment can intersect if the become adjacent in the sweep data structure
- To detect intersections we want to check each adjacent pair at the first event point where they became adjacent.
    - we insert a segment at a L endpoint
        - check if it intersects with its new neighbors
    - we remove a segment at a R endpoint
        - check if the removed segment's neighbors intersect.

Segment Intersection

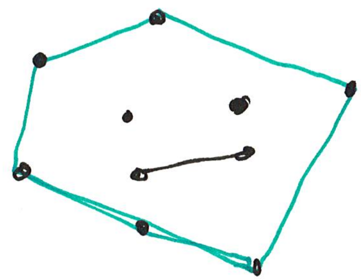
- we encounter $2n$ endpoints
  - the sweep data structure need $O(\log n)$ insertions and deletions in order for us to get $O(n \log n)$
  - any self-balancing tree works for this
    - red-black tree
    - AVL tree

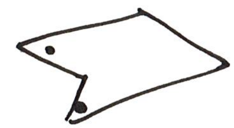
Complexity

$\Theta(n \log n)$



Convex Hull



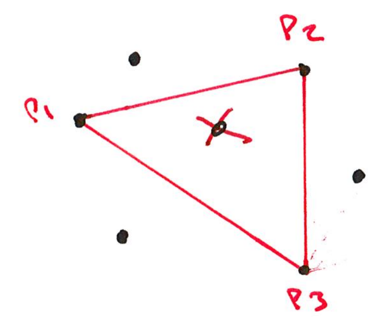


- Given a set of n points S, find the smallest convex polygon containing all of the points.

- we'll present the convex polygon as the ordered sequence of vertices

<u>Brute Force</u> $O(n^4)$

- any point inside any triangle is not part of the hull



- sort the ~~remaing~~ remaining points by polar angle around some centroid

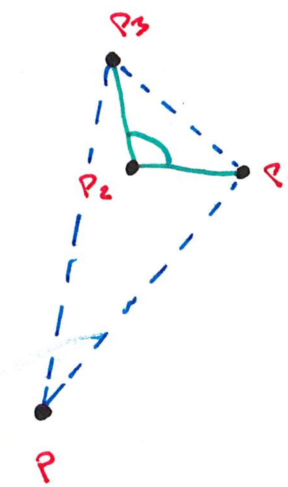
<u>Option 2</u> $O(n^3)$

- for each pair of points p_1, p_2
    - check if all other points are on the same side of the line $\overleftrightarrow{p_1 p_2}$

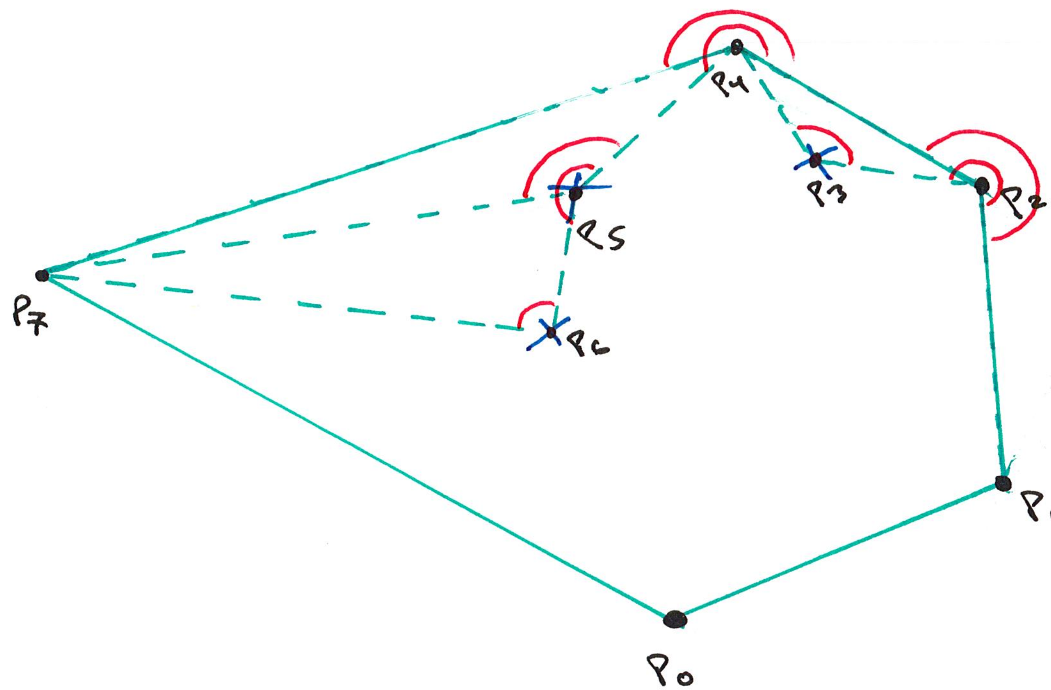
Graham's Scan

1. find a known point on the hull p
2. Sort the remaining points by polar angle in counter clockwise order around p
3. Process the sorted points in consecutive triples $p_1 p_2 p_3$
   - if $\angle p_1 p_2 p_3 < \pi$ then p_2 is in the interior of the triangle $p p_1 p_3$ and therefore not part of the hull.

     Discard p_2 and try $p_0 p_1 p_3$

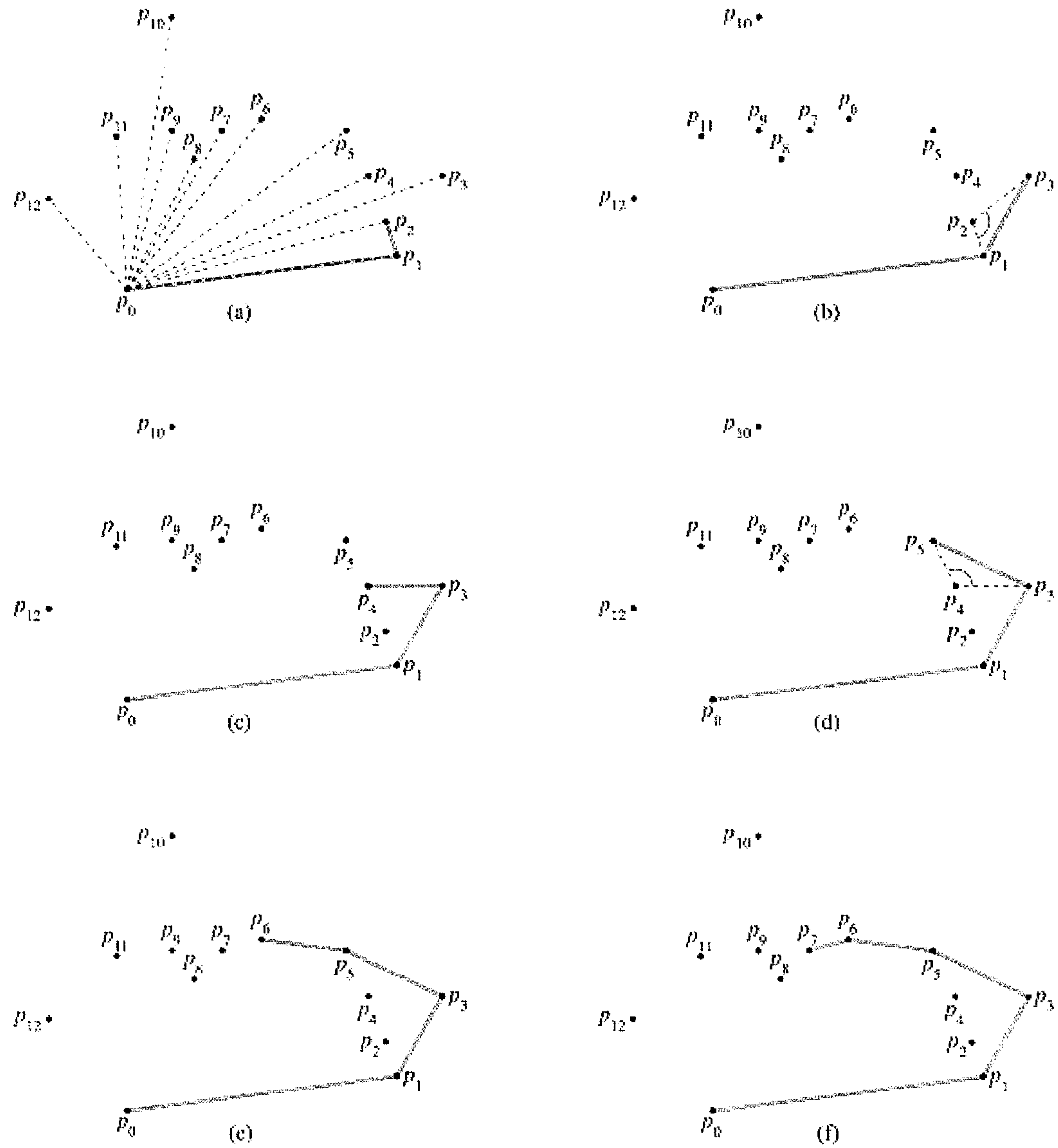
     

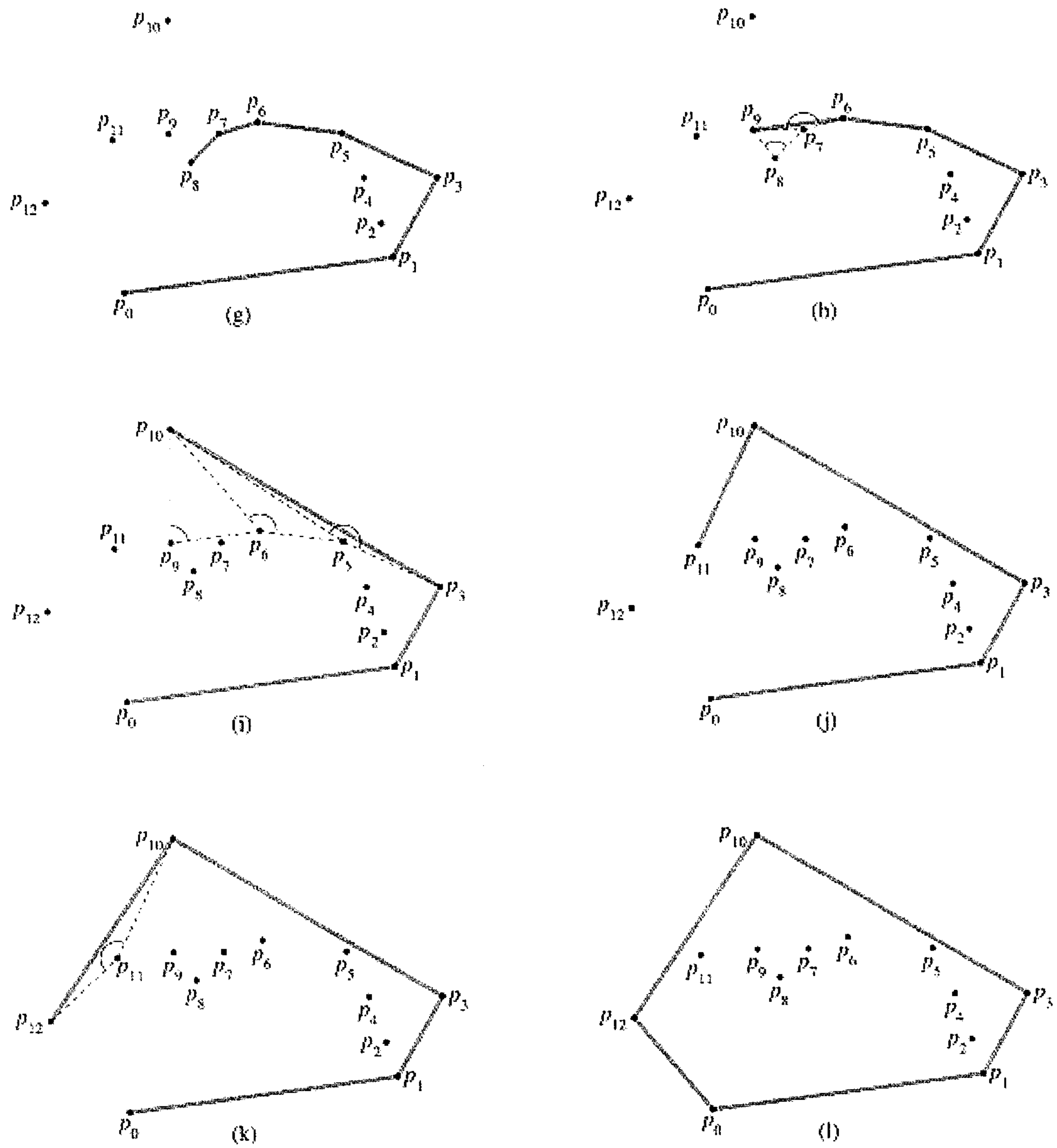
   - otherwise advance to $p_2 p_3 p_4$


P4
P3
P2
P5
P7
P6
P1
P0

GRAHAMSCAN(Q)

let p_0 be the point in Q with the lowest y-coordinate $\Theta(n)$
let $\langle p_1, p_2, \ldots, p_m \rangle$ be the remaining points in Q $\Theta(n \log n)$
sorted by polar angle around p_0
if $m < 2$
    return "convex hull is empty"
else let S be an empty stack
    PUSH(p_0, S)
    PUSH(p_1, S)
    PUSH(p_2, S)
    for $i = 3$ **to** m
        while the angle formed by p_i and the top 2 elements
        of S makes a non-left turn $\Theta(n)$
            POP(S)
        PUSH(p_i, S)
return S


(a)
(b)
(c)
(d)
(e)
(f)



(g) (h) (i) (j) (k) (l)