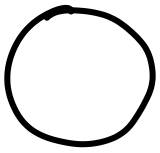
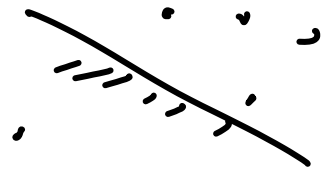
Non-orthogonal query rectangles

These are more difficult and the algorithms for k-d trees/range trees don't work

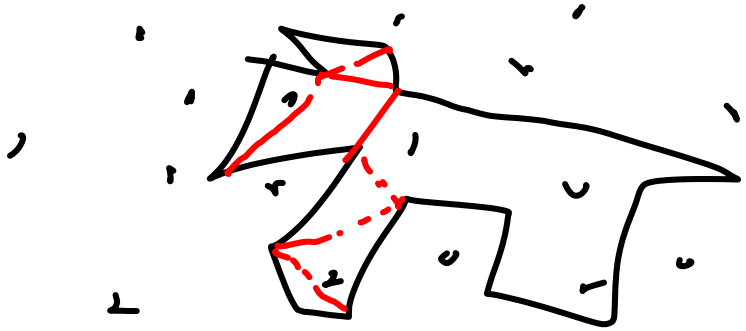
- Circular range query



- Half-plane range query



- Polygonal range query

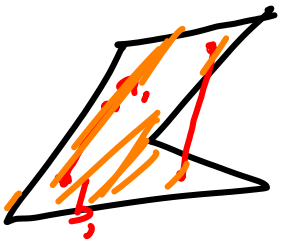


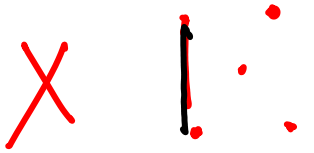
(triangular range query)

Convex hull

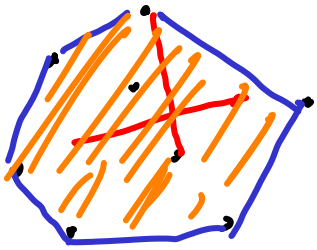
Defn: Given a set of points P on a plane, the smallest convex set that contains P is called the convex hull of P and denoted by $CH(P)$

Convex set: Given any two points a, b in the set, the segment $\overline{ab}$ is completely contained in the set





Discrete set of points cannot be convex

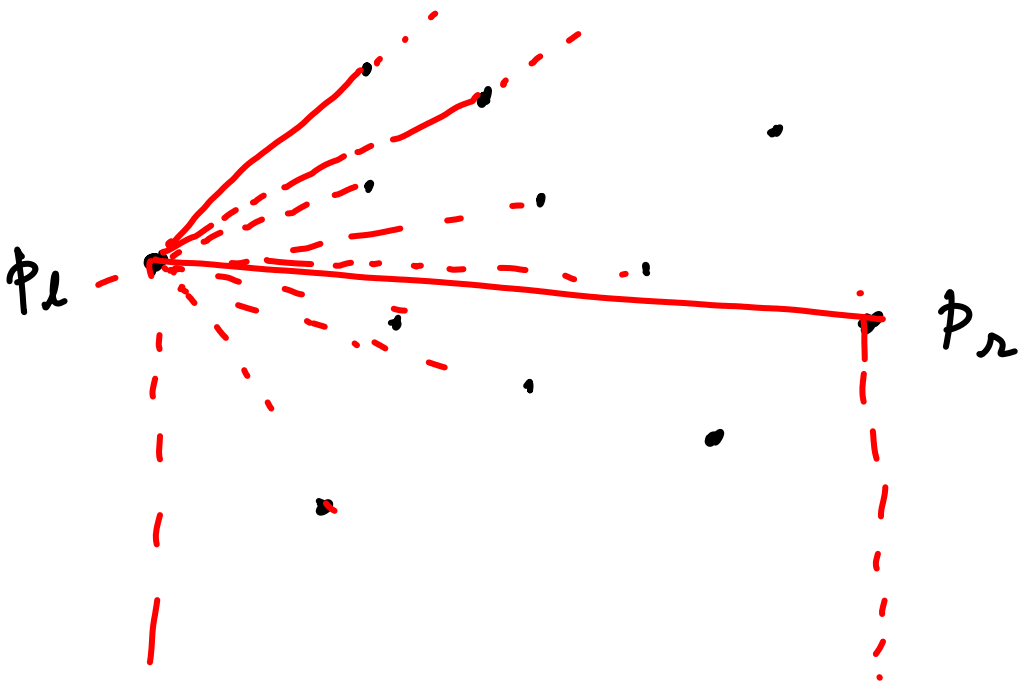


it is a closed set

$CH(P)$ where P is a set of n points on plane

It is characterized by "boundary points" that belong to P and a straight line segments joining the boundary points

Given a set P of n points on the plane, we want to construct a convex hull $CH(P)$ - we want find the ordered set of boundary points



We are discovering the boundary points one after the other in a clockwise ordering from p_l

The running time is proportional to

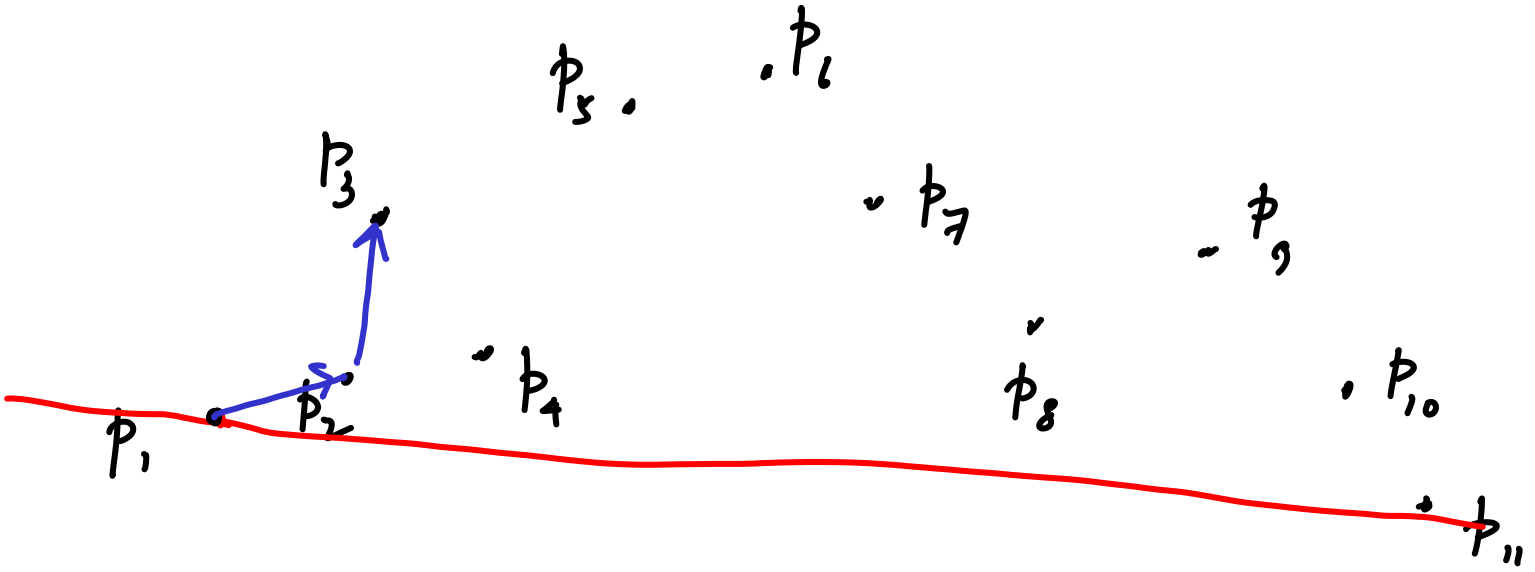
boundary points $\times$ time to find the next bd point

Jarvis March

$\leq O(n^2)$ $\qquad$ $O(n)$

Can we design faster algorithm

Step 1 Sort the points according x axis



Step 2 Visit the points in increasing x-coordinates

If last-three consecutive points p_i, p_{i-1}, p_{i-2} is a right turn then we move ahead

else we drop p_{i-1} from further consideration

By using a stack the entire algorithm runs in $O(n)$ -time (after sorting)

Graham Scan