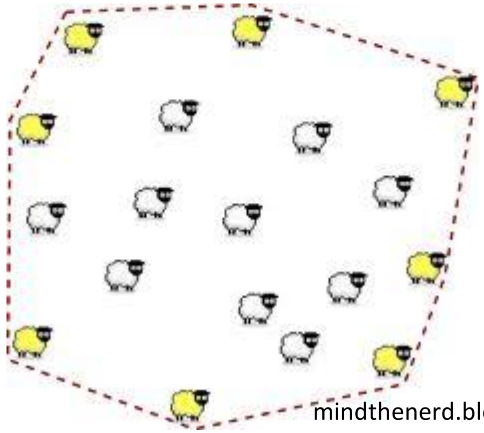
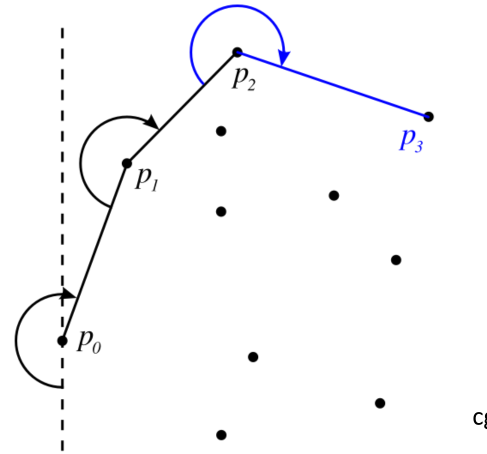
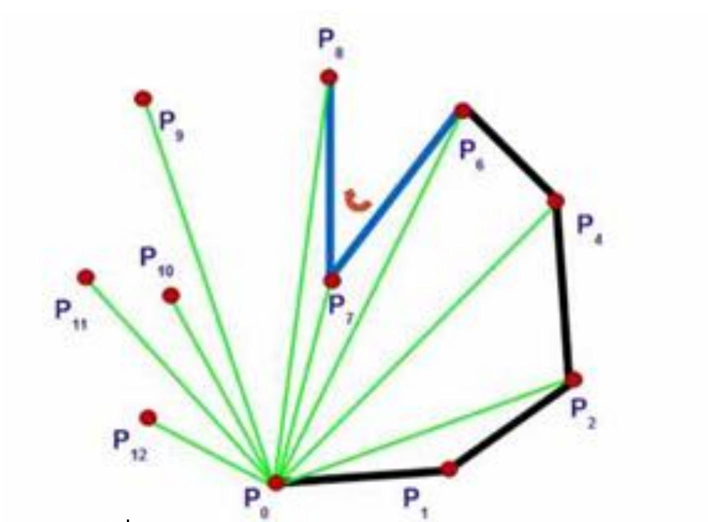

mindthenerd.blogspot.com

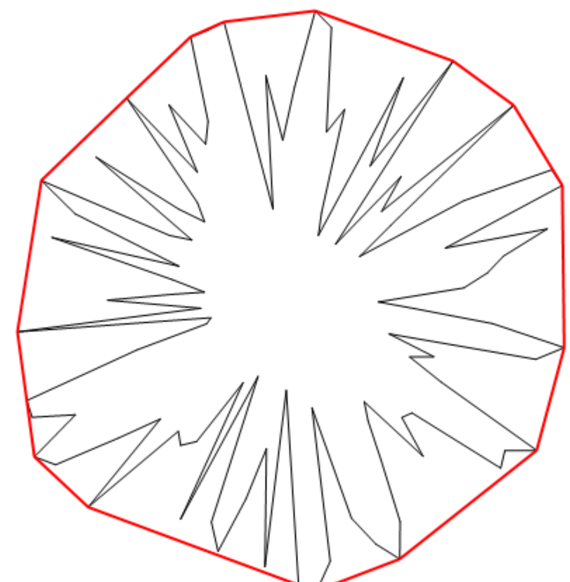


cgi.di.uoa.gr

# Convex hull in 2D



www.youtube.com

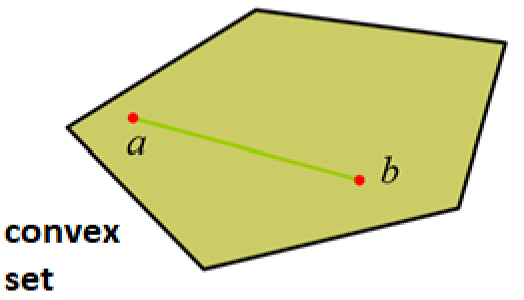

www.codeproject.com

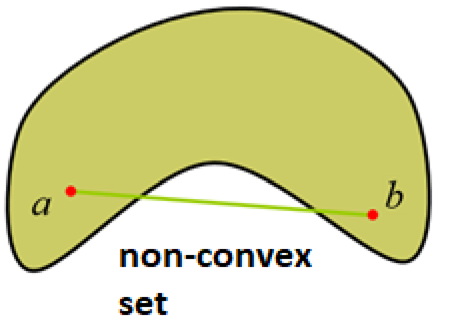
# Before we start …

- Link to study materials of Geometric algorithms course
- https://is.muni.cz/auth/do/sci/UMS/el/geometricke-alg/index.html

# Convex hull

- Set M is **convex** if a line segment connecting its two arbitrary points fully lies inside M

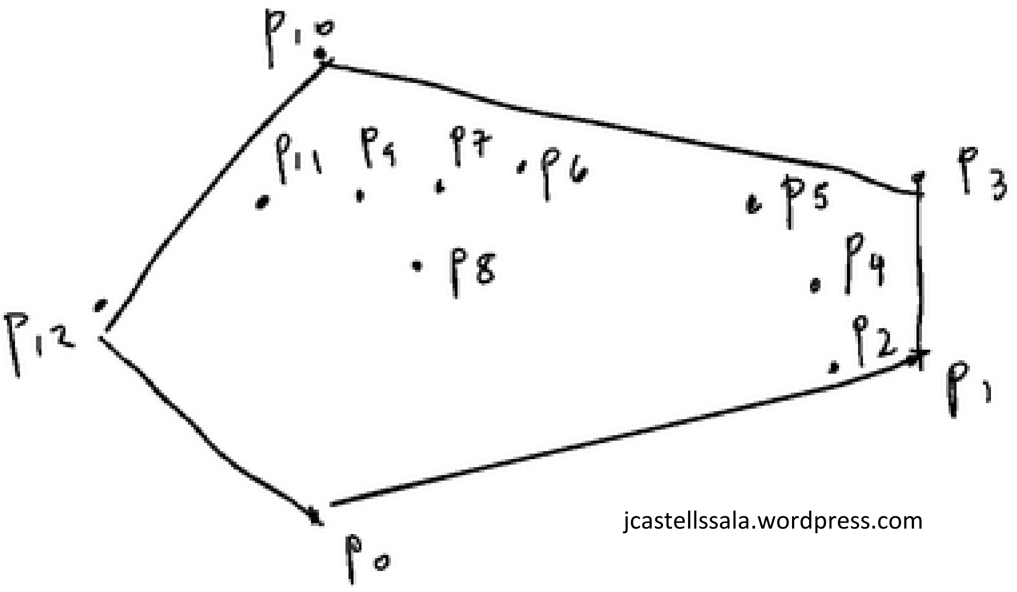



- **Convex hull** of a set of points X in the Euclidean space corresponds to the smallest convex set containing X

# Convex hull

- Input: *n* points on a plane



# Convex hull

- Convex hull in 2D:

  = convex polygon

  - Represented by an **ordered sequence of vertices** (counter clockwise)

- Convex hull in 3D:

  = convex polyhedron

  - Represented by a **planar graph**

# Convex hull – algorithms

- Gift Wrapping (Jarvis March)
- Graham Scan
- Incremental algorithm
- Divide and conquer

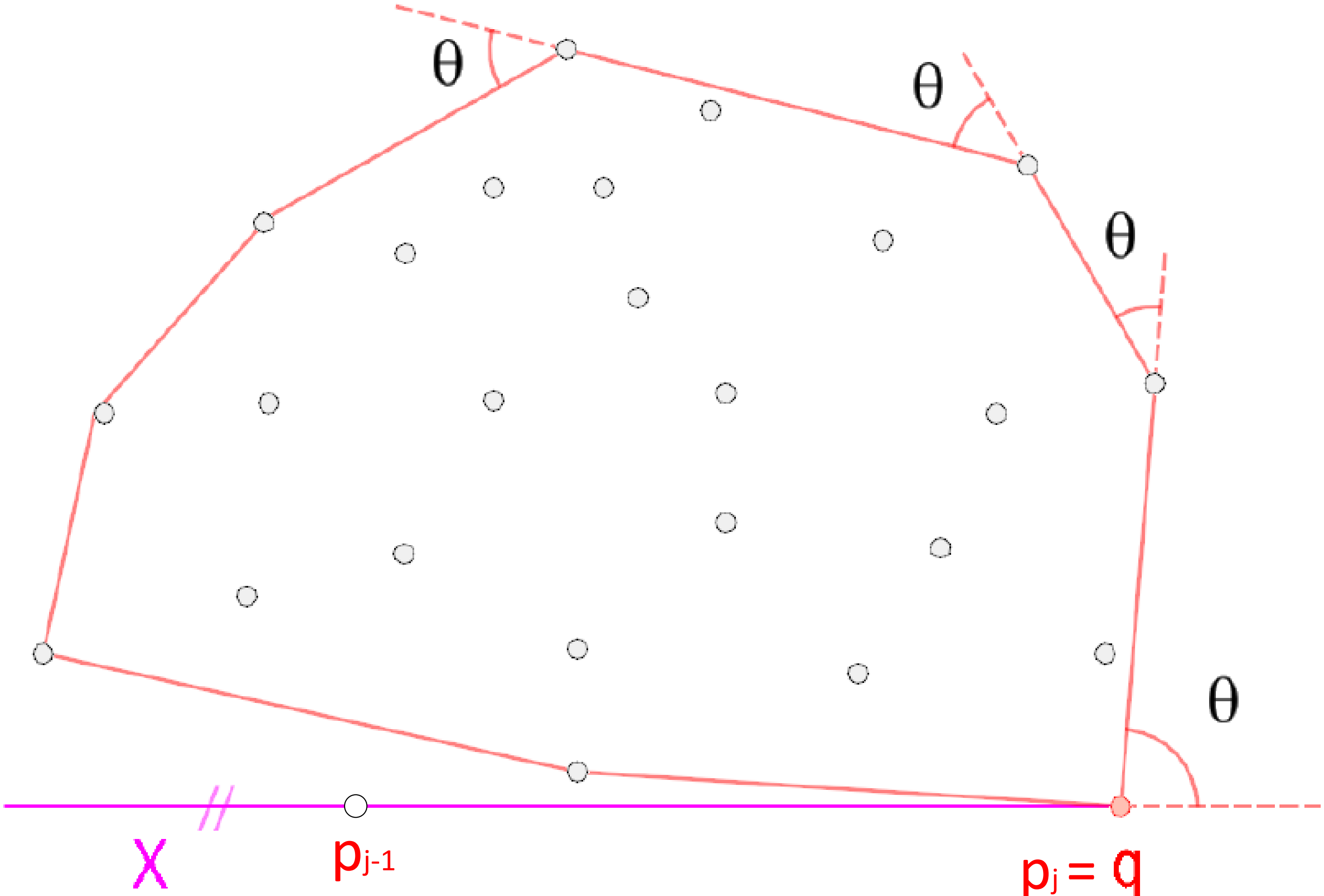
# Gift wrapping (Jarvis March)

- Resembles wrapping gifts, proposed by Jarvis (1973)
- Simple implementation and extension to 3D
- Assumption: set X does not contain three co-linear points
- Complexity: Preprocessing $O(n)$, algorithm $O(n^2)$

# Gift wrapping (Jarvis March)

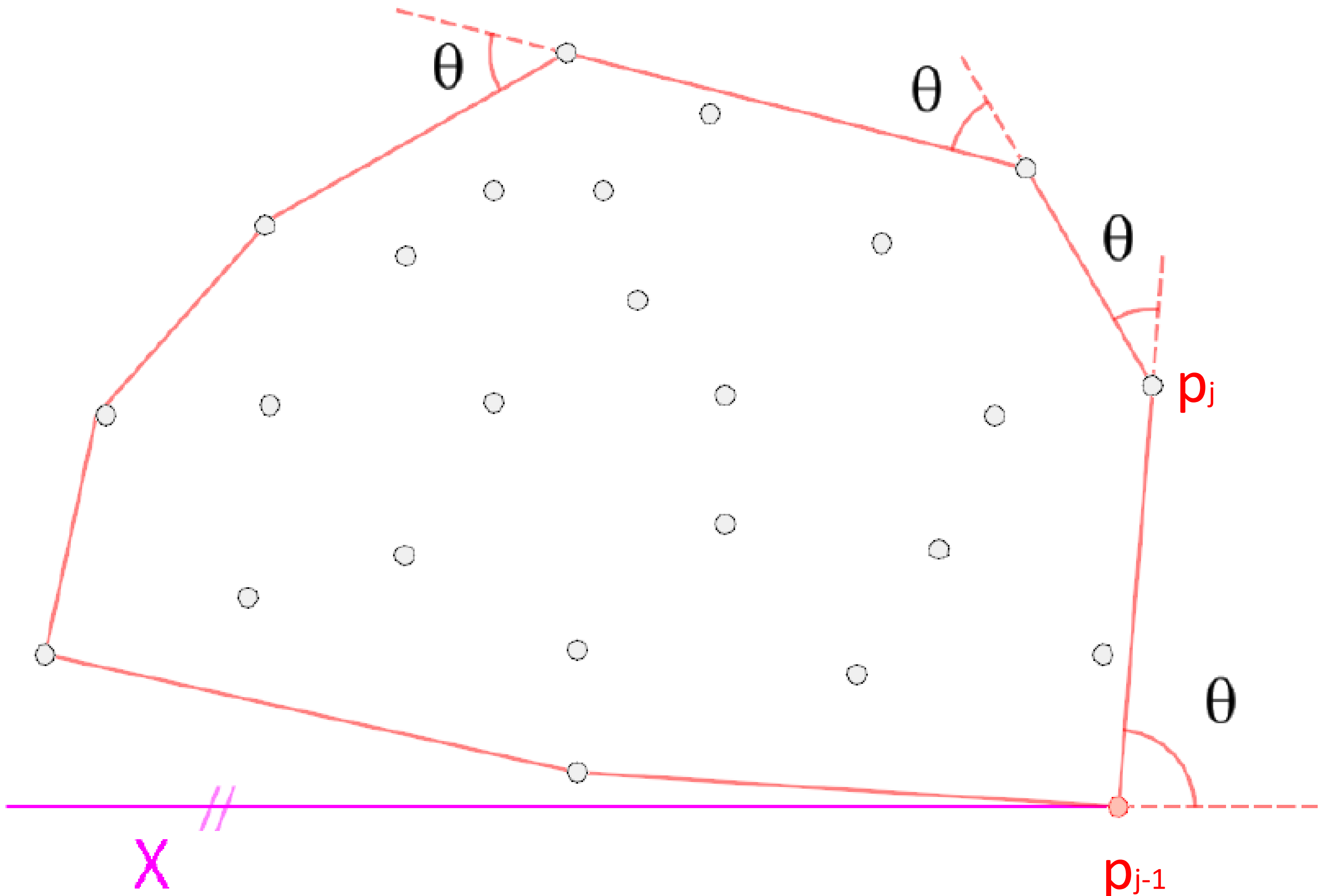
- **Principle**:
  - Find pivot $q$ ($q$ = max($y_i$))
  - Add q to the convex hull $H$
  - $p_{j-1}$ = arbitrary point on x axis, $p_j = q$, $p_i = p_{j-1}$
  - Repeat until $p_i \neq q$:
    - Repeat $\forall p_i \notin H$ and points $p_{j-1}$, $p_j$ :
      - Find $p_i$ for that the angle $\Theta = \min(\Theta_i)$
    - Add $p_i$ to $H$
    - $p_{j-1} = p_j$, $p_j = p_i$

# Gift wrapping (Jarvis March)




http://web.natur.cuni.cz/~bayertom/Adk/adk4.pdf

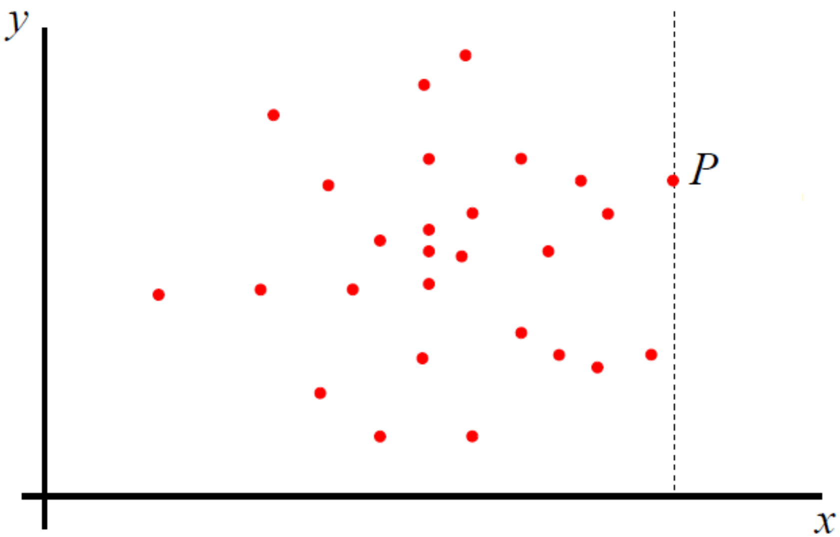
# Gift wrapping (Jarvis March)




http://web.natur.cuni.cz/~bayertom/Adk/adk4.pdf

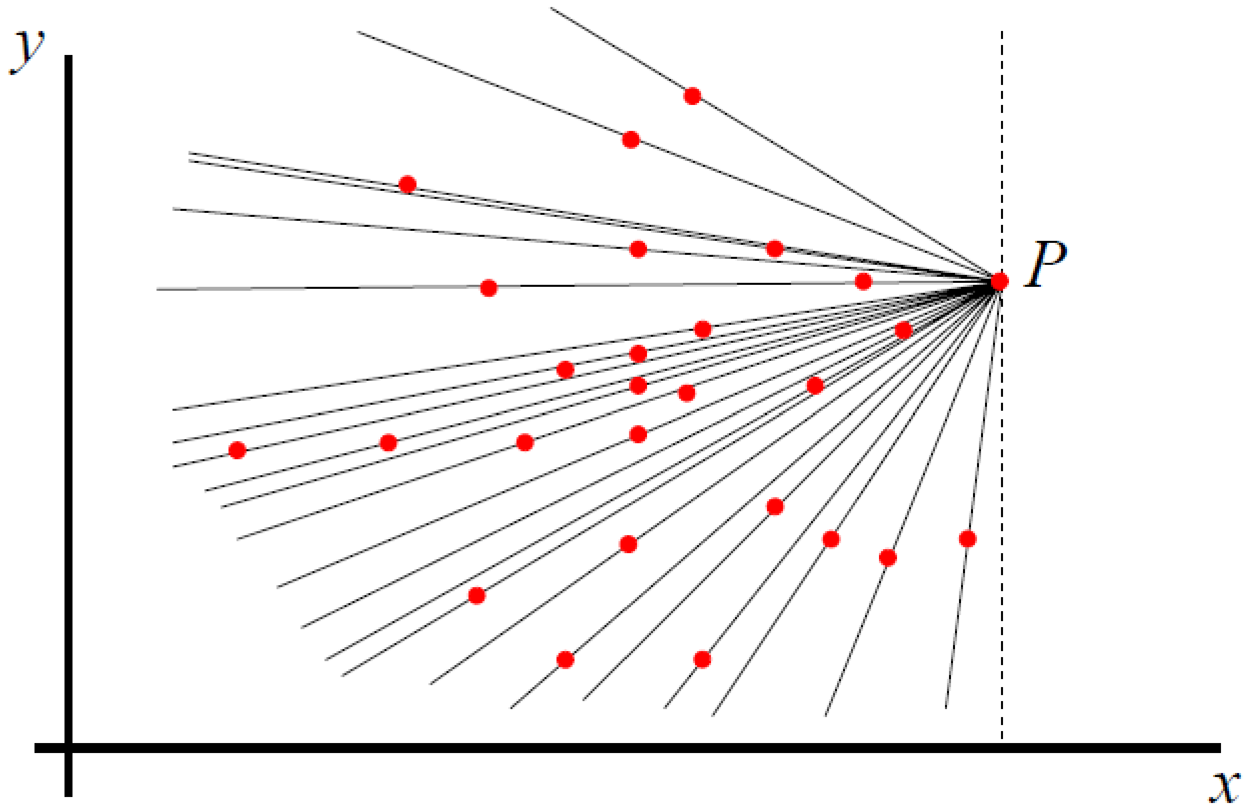
# Implementation

- We find point *P* with the highest *x*-axis value – this is one of the vertices of the convex hull
- In this point *P* we determine so called **separating line** (often parallel to *y* axis). All points in the input set lie in the same half-plane, determined by the separating line



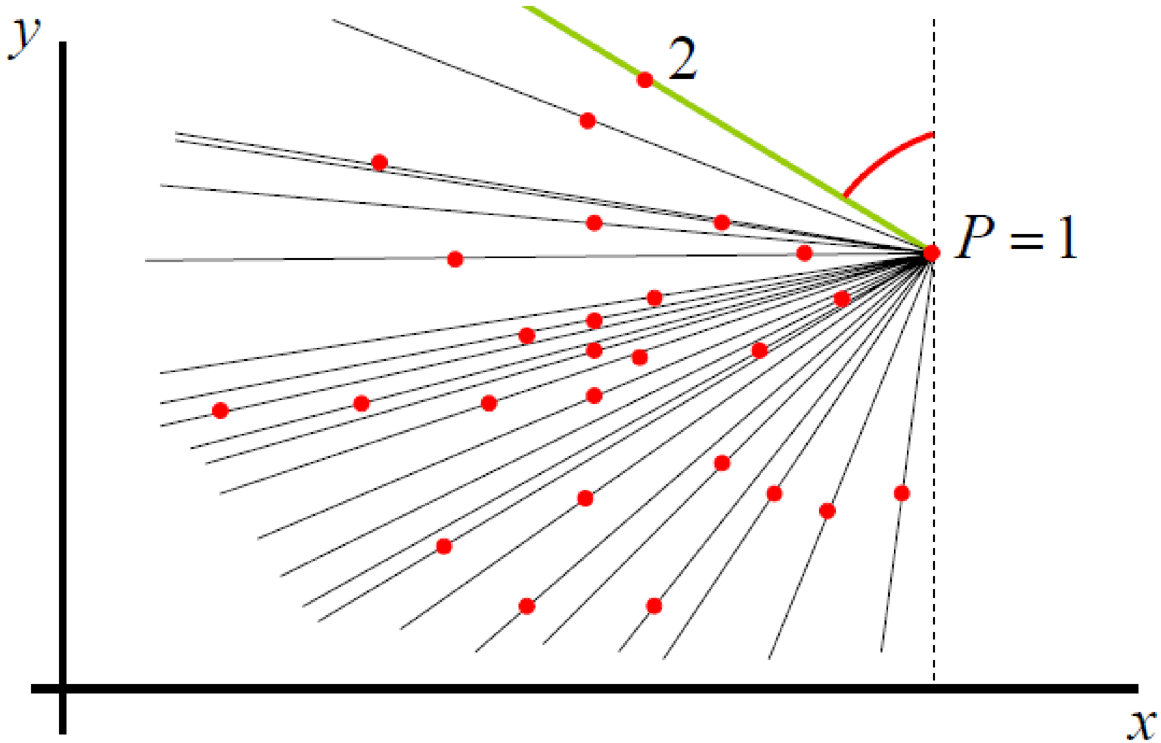
# Implementation

- From P we shoot rays heading to all other points of the input set



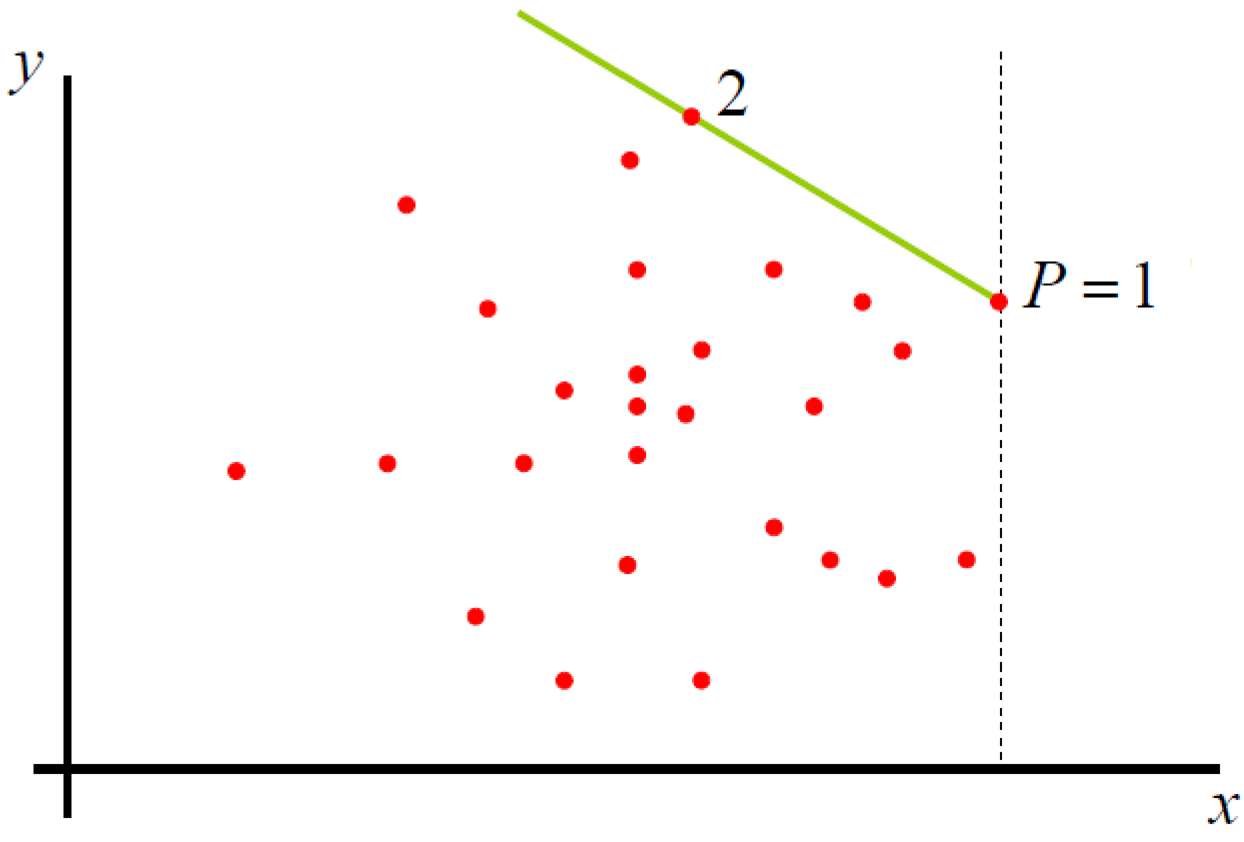
# Implementation

- We select a ray which has the minimal angle with the first (separating) line. We have next vertext of the convex hull (*2*)



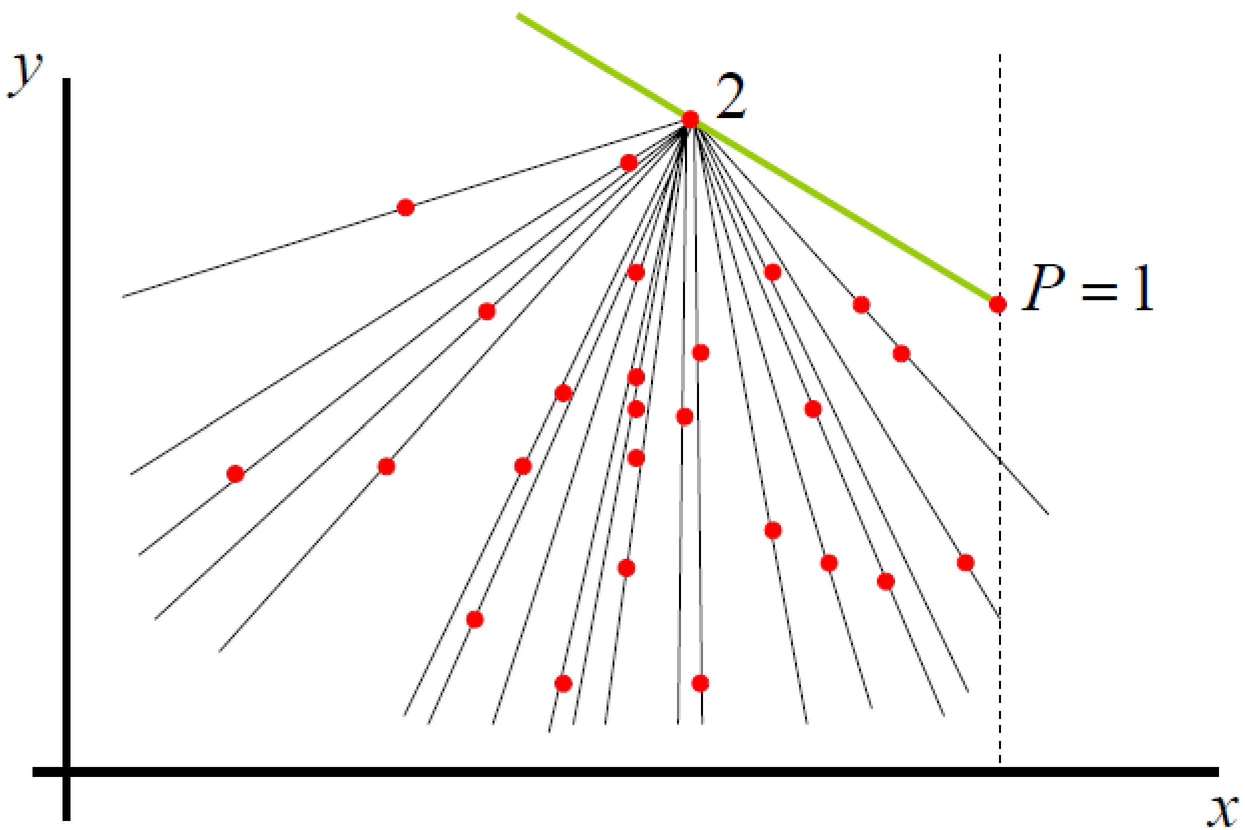
# Implementation

- New edge of the convex hull is *1-2*



# Implementation

- Repeat this until we will reach the first point P again



# Implementation