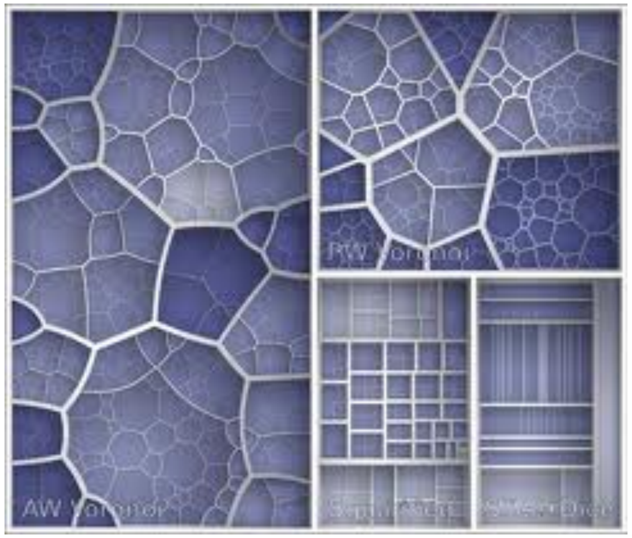
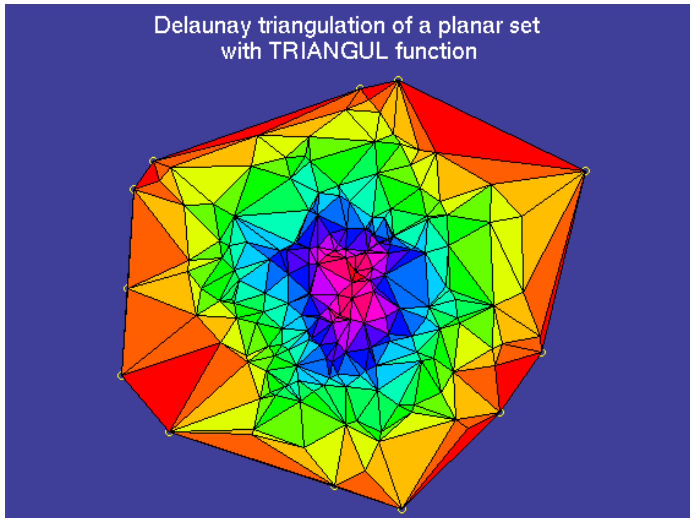


rickosborne.org

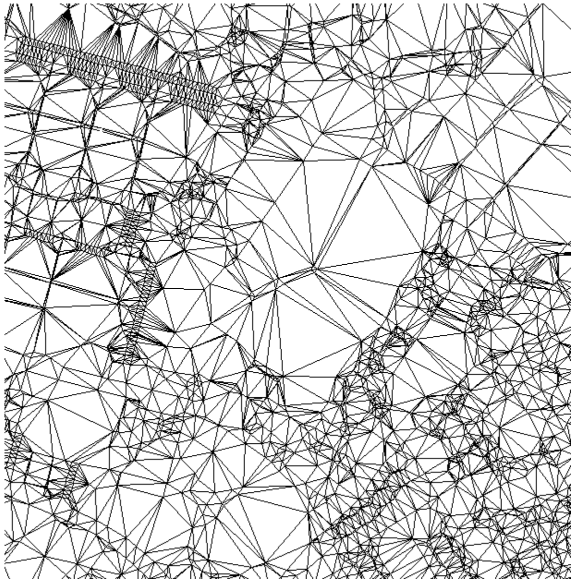


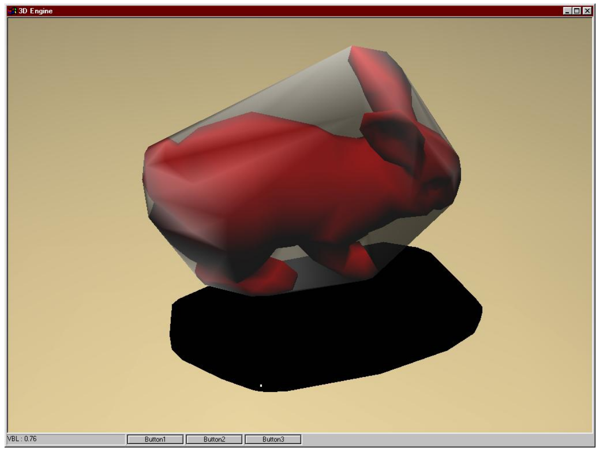
puddle.mit.edu

PA093
Computational geometry project

Barbora Kozlíková
xkozlik@fi.muni.cz

Pavol Ulbrich
410146@mail.muni.cz


herakles.zcu.cz



www.codercorner.com

Introduction

- Course taught every second week, starting with the first week of semester
- Communication channel:
  - Primary by mail: xkozlik@fi.muni.cz, 410146@mail.muni.cz
  - Personally after arranging an appointment (via mail ☺️)

Introduction

- Course connected to M7130 Computational geometry
- https://is.muni.cz/auth/do/sci/UMS/el/geometricke-alg/index.html
- Implementation of selected algorithms
- Java, C++, Processing, ...?
- Credits given after submitting all assignments

Course outline

- Everything will be implemented into a basic framework – it will be a base for all assignments
  - Convex hull in 2D
  - Triangulation
  - k-D trees
  - Voronoi diagrams
  - …

Outline for the next weeks …

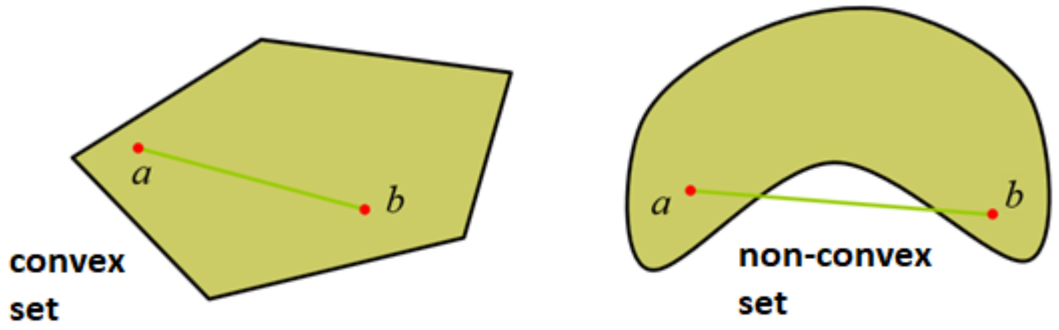
- 18. 9. – intro, basic framework, convex hull
- From 2. 10.
  - Convex hull continued
  - Triangulation of points in a plane
  - k-D tree
  - Delaunay triangulation
  - Voronoi diagram
  - Christmas ☺️

TASK 1 - Basic framework

- It should contain:
  - Visualization (rendering) window
  - Random points generation
  - Add point on mouse click
  - Delete point on mouse click
  - Move point using mouse move
  - Clear scene
- You will start to work on that right now…

Convex hull

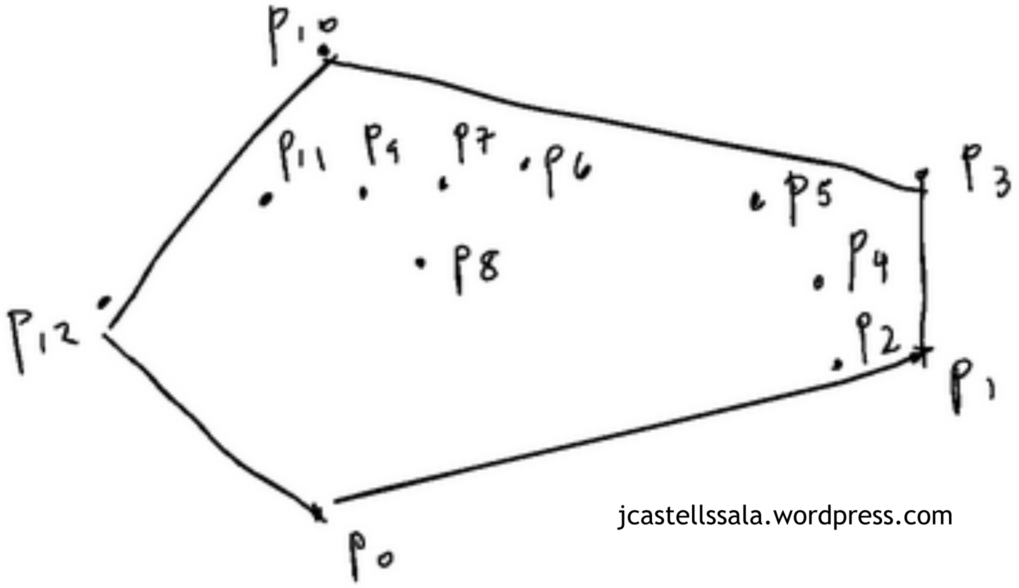
- Set M is **convex** if a line segment connecting its two arbitrary points fully lies inside M



- **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



jcastellssala.wordpress.com

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

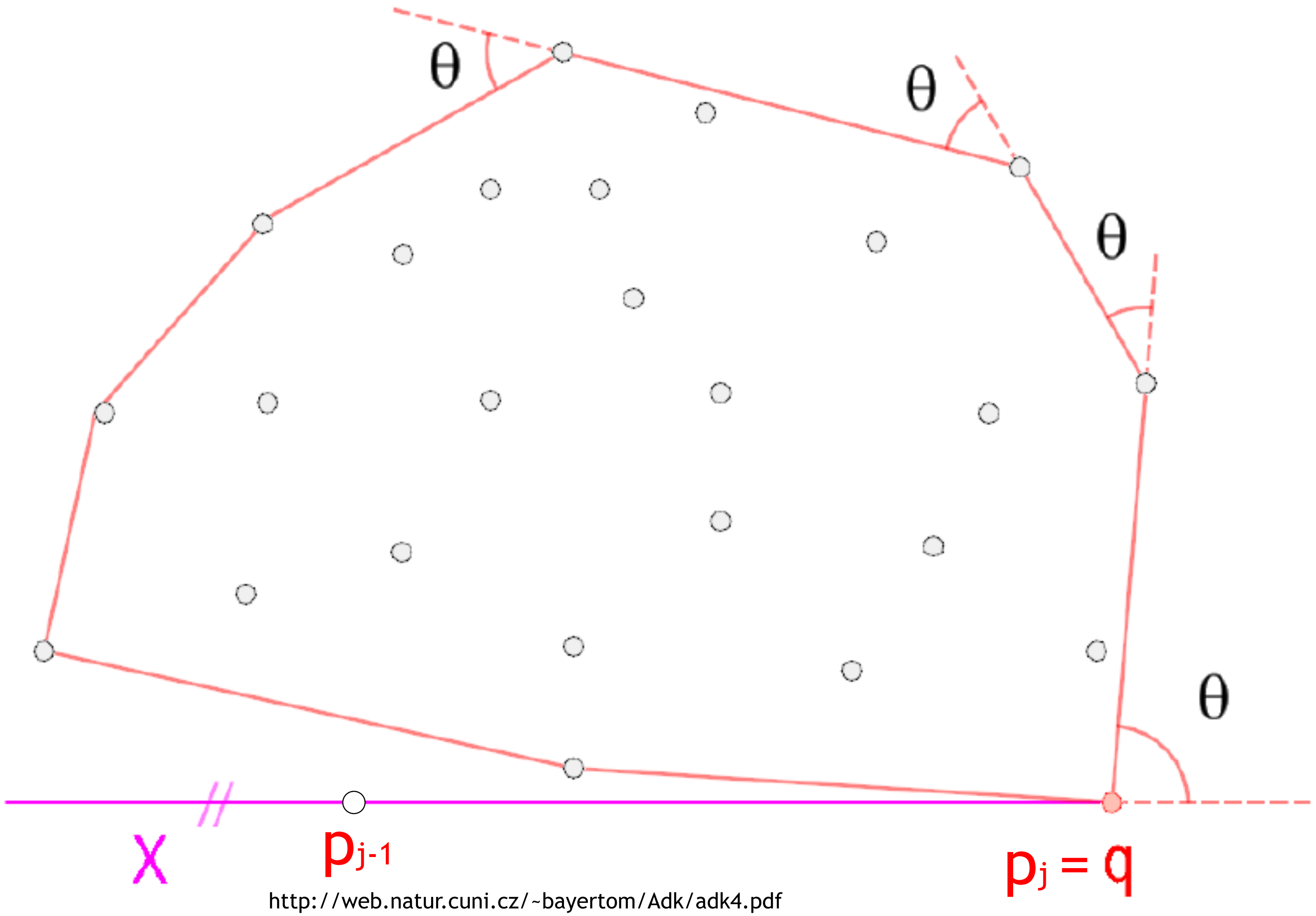
TASK 2 - Gift wrapping (Jarvis March)

- 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)

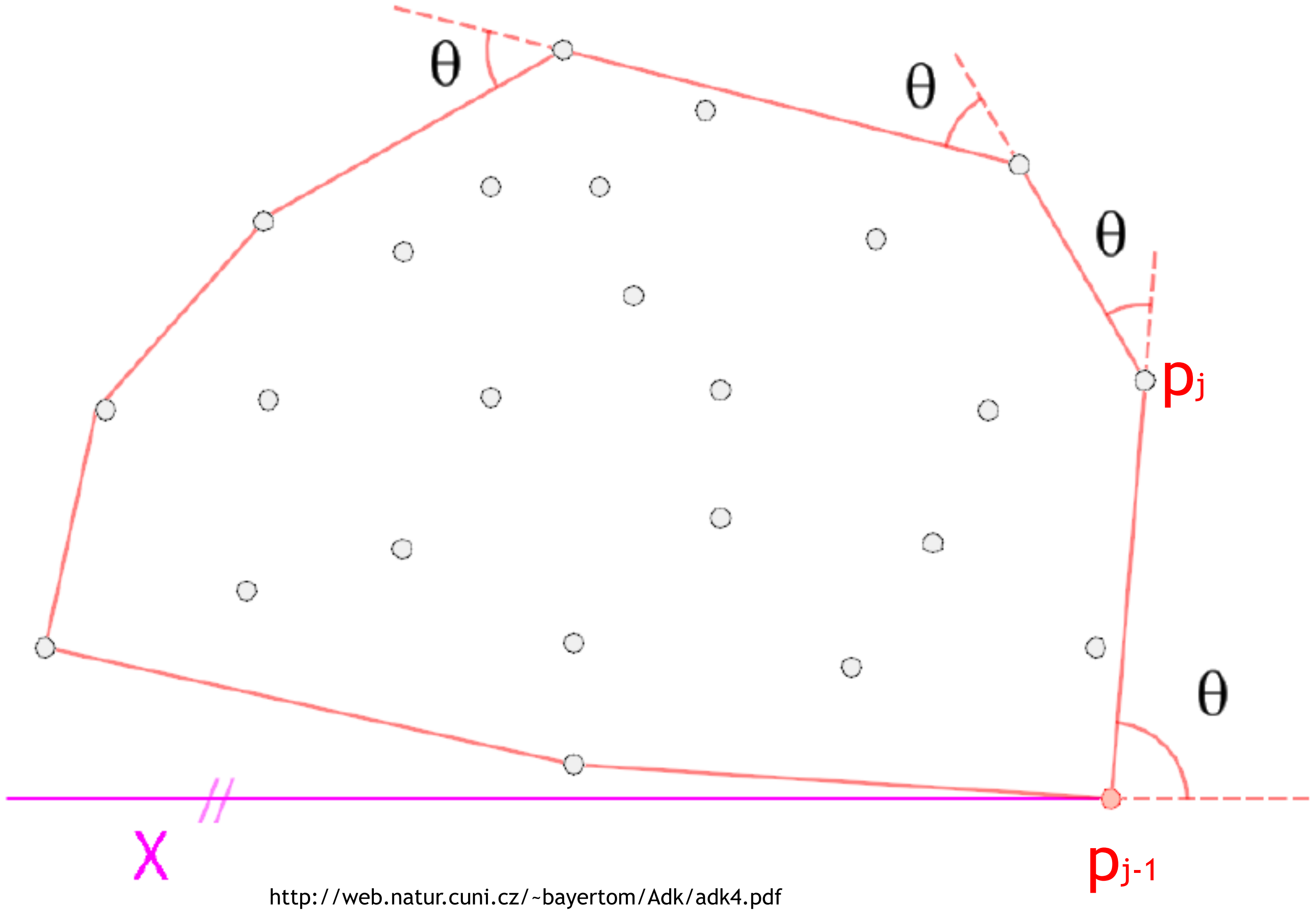
- **Principle:**
  - Find pivot q (q = max(y_i))
  - Add q to the convex hull H
  - p_{j-1} = arbitrary point on x axis, $p_j = q$, $p_i = p_{j-1}$
  - Repeat until $p_i \neq q$:
    - Repeat $\forall p_i \notin H$ and points p_{j-1}, p_j :
      - Find p_i for that the angle Θ = min(Θ_i)
    - Add p_i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

Gift wrapping (Jarvis March)



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

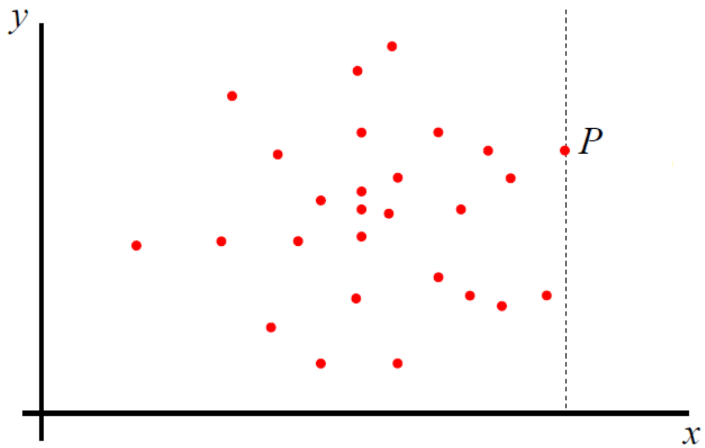
Gift wrapping (Jarvis March)



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

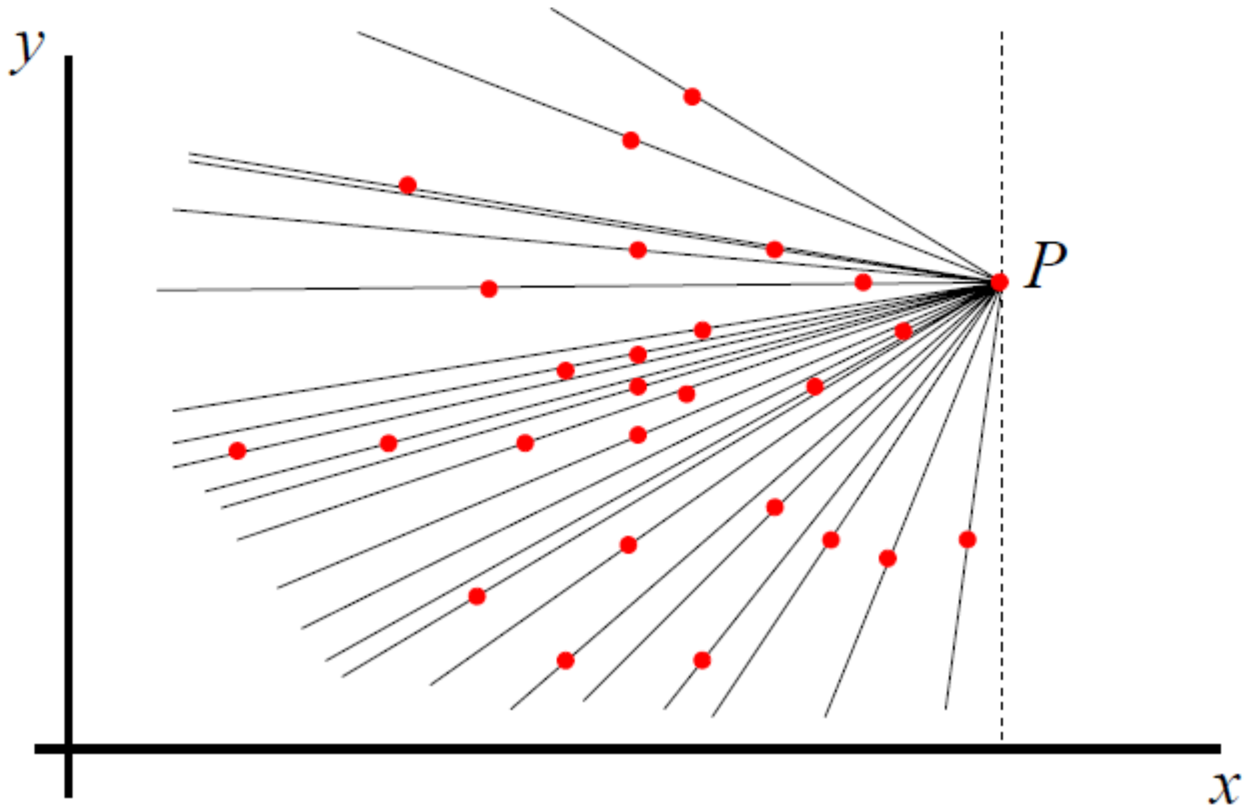
Implementation

- 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



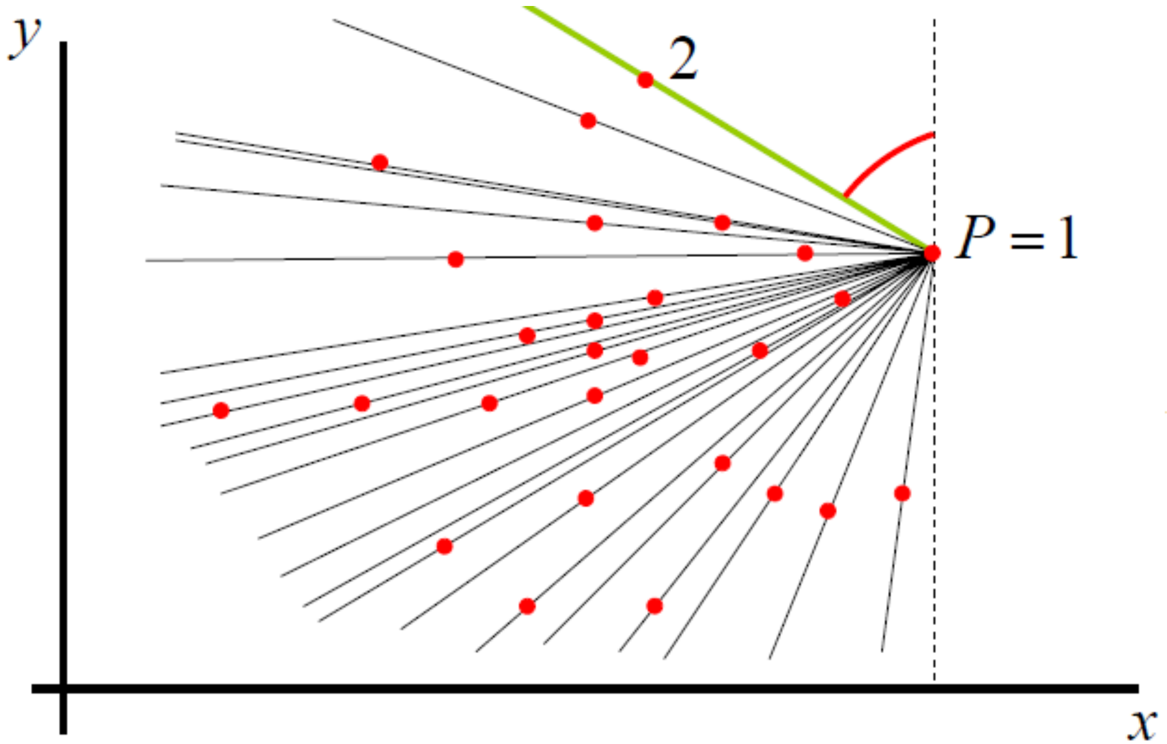
Implementation

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



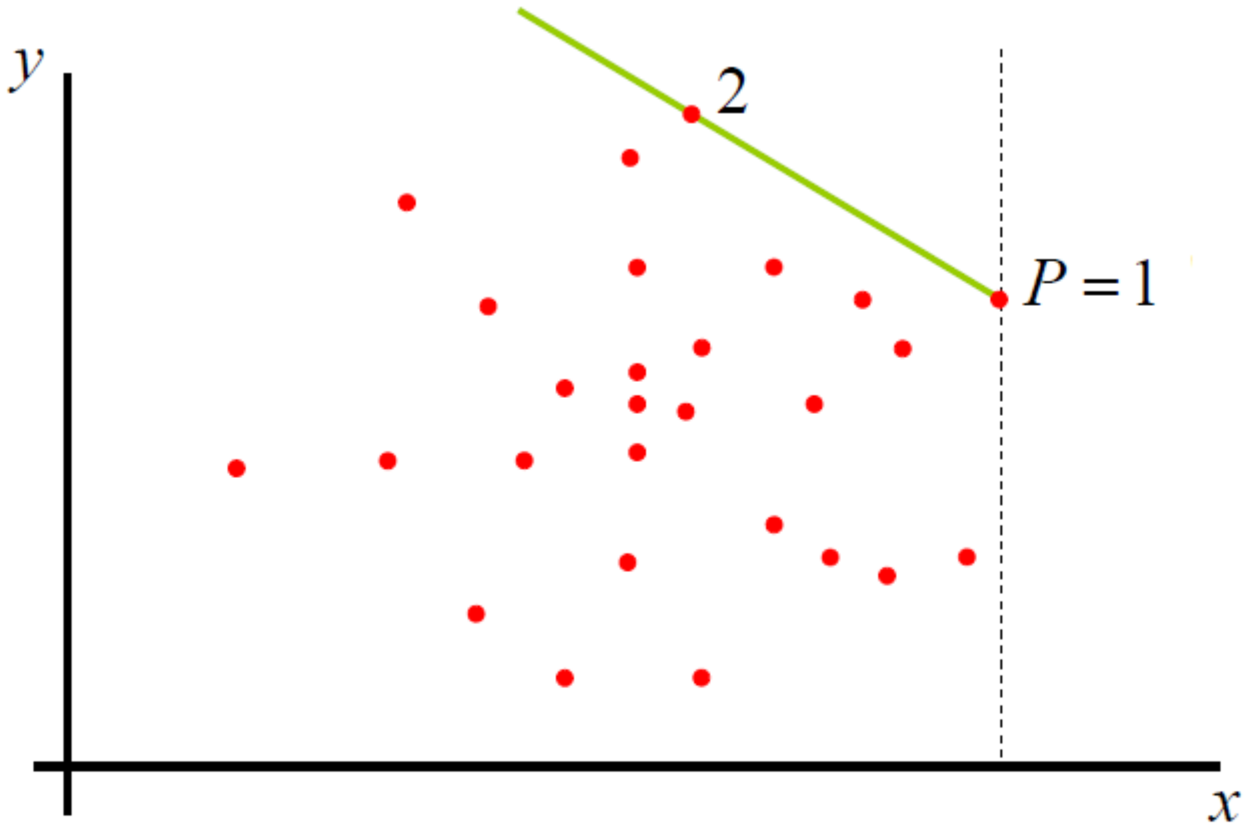
Implementation

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



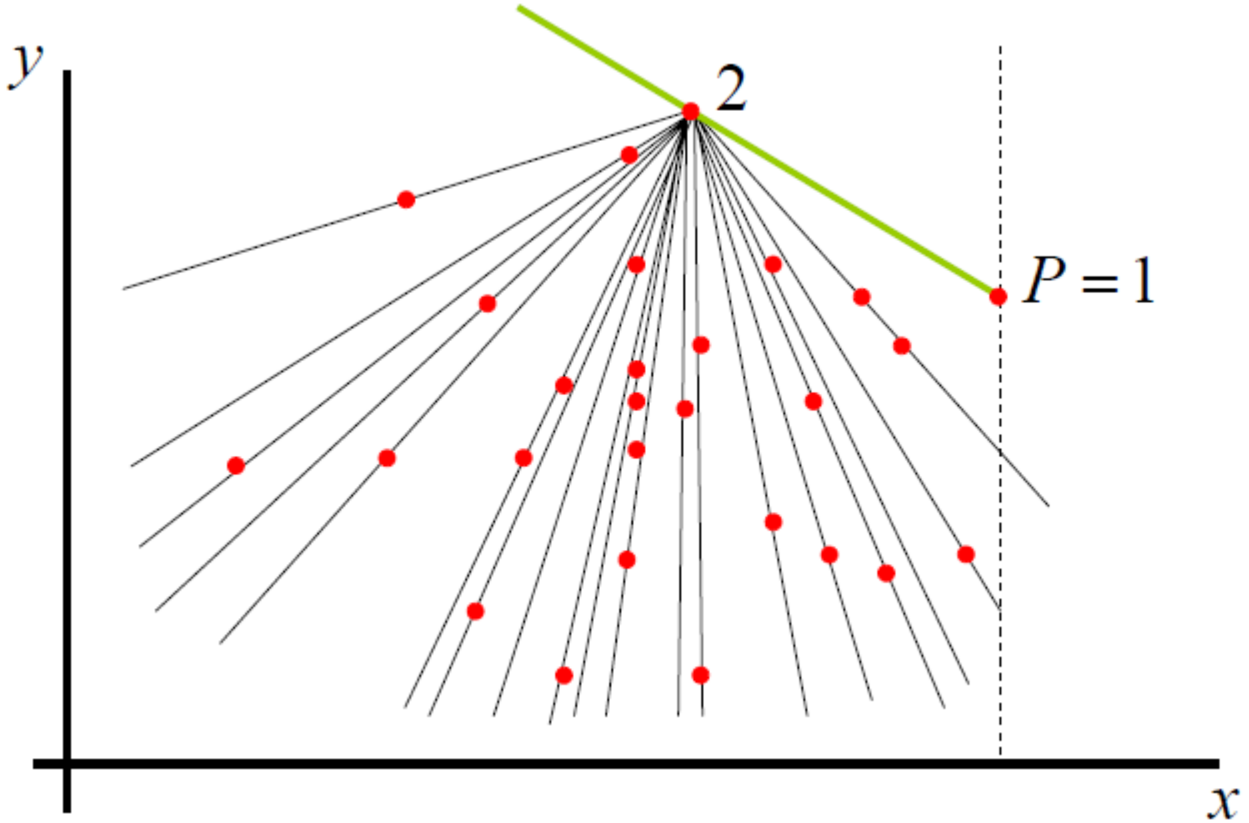
Implementation

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

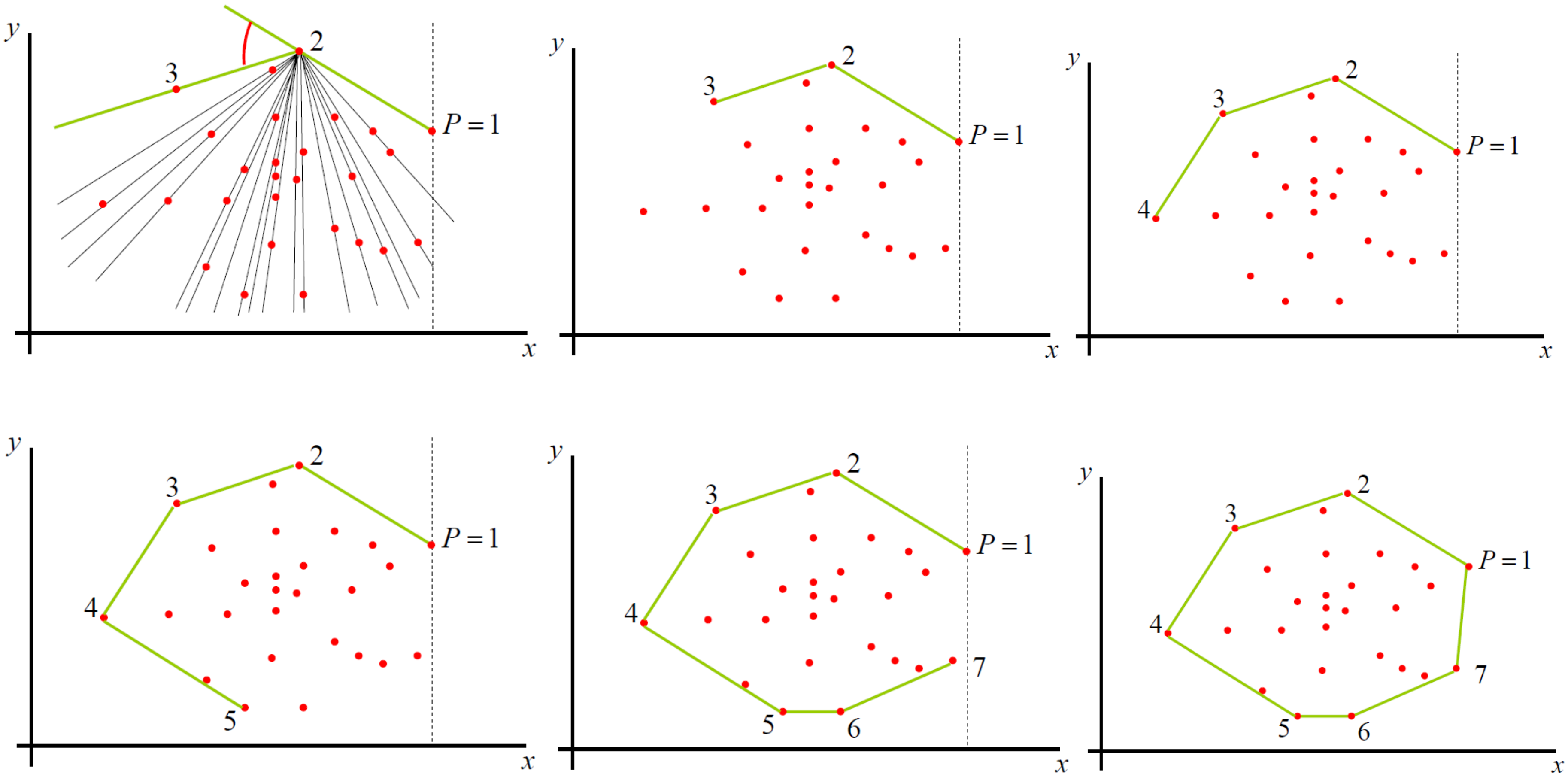


Implementation

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



Implementation



TASK 2

- Take your basic framework and implement the Gift Wrapping algorithm into that
  - Input = set of points added by mouse clicking
  - Output = visualized convex hull
  - Implement also an update of the convex hull when the user moves the points in the scene (one of the required functionalities for the basic framework). This update does not have to be instant (but can be ☺️), can be performed by clicking on a button …