Computational Geometry
Chapter 2: Convex Hull

Prof. Dr. Sándor Fekete

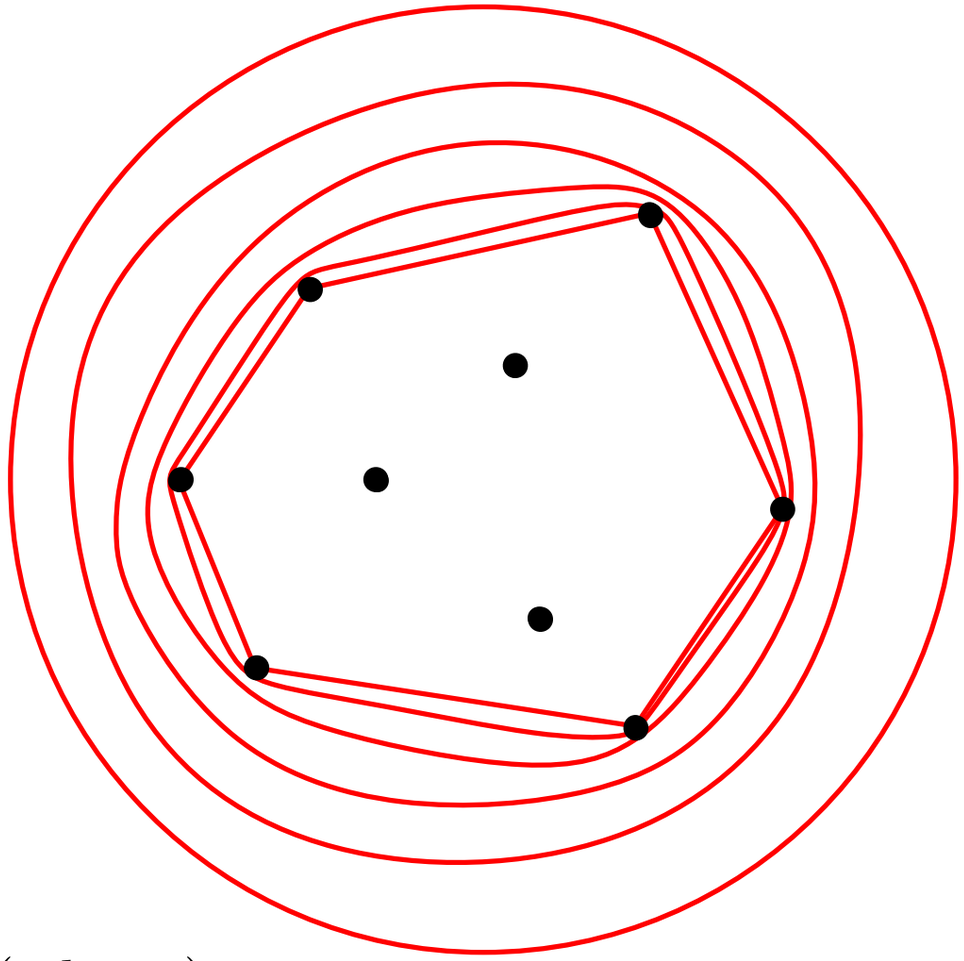
Algorithms Division
Department of Computer Science
TU Braunschweig

Technische Universität Braunschweig

1. Introduction and Definitions
2. Interlude: Algorithmic Paradigms
3. Jarvis' March
4. Quickhull
5. Divide-and-conquer and incremental construction
6. Graham's Scan
7. Optimal output-sensitive construction

Task:

- Given: Set of n points in $\mathbb{R}^d$
- Wanted: Smallest enclosing convex object



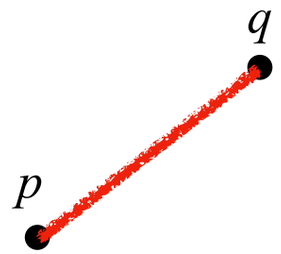
Intuition in $\mathbb{R}^2$:

- Draw points on a wooden board.
- Put in nails at points.
- Let a rubber band snap to the nails.

Theorem 2.18: Computing the convex hull takes $\Omega(n \log n)$.

Theorem 2.35: Computing the h vertices of the convex hull can be done in $O(n \log h)$.

Convex Hulk



Definition 2.1

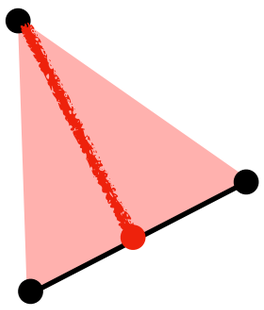
For $p, q \in \mathbb{R}^d : \overline{pq} := \{x \in \mathbb{R}^d \mid \exists \alpha, \beta \in \mathbb{R}, \alpha, \beta \geq 0, \alpha + \beta = 1, x = \alpha p + \beta q\}$

$p + \lambda(q - p), 0 \leq \lambda \leq 1$
$= (1 - \lambda)p + \lambda q$

Definition 2.2

For $\{p_0, \ldots, p_{n-1}\} \subset \mathbb{R}^d$ point $x \in \mathbb{R}^d$ is a convex combination of $\{p_0, \ldots, p_{n-1}\}$, if

$\exists \alpha_0, \ldots, \alpha_{n-1} \in [0, 1]$ with 1. $\sum_{i=0}^{n} \alpha_i p_i = x$



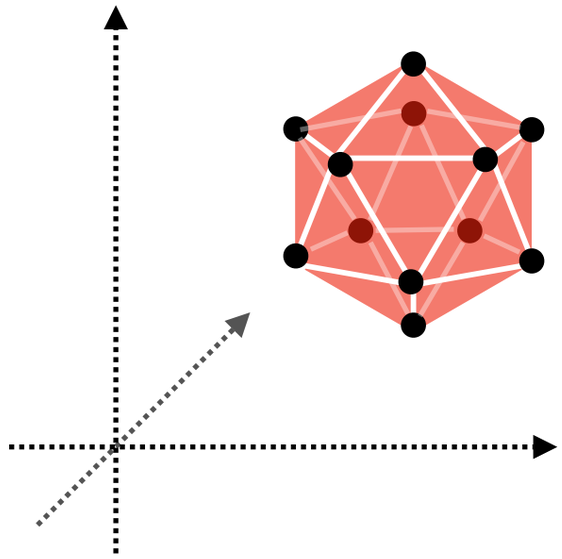
2. $\sum_{i=0}^{n} \alpha_i = 1$

- $\overline{pq} = \{x \mid x$ convex combination of $\{p, q\}\}$
- $\triangle(p, q, r) = \{x \mid x$ convex combination of $\{p, q, r\}\}$

Definition 2.3

Convex hull $conv(\mathcal{P})$ of $\mathcal{P} := \{p_0, \ldots, p_{n-1}\}$:

$$conv(\mathcal{P}) := \{x \in \mathbb{R}^d \mid x \quad \text{convex combination of} \quad \mathcal{P}\}$$

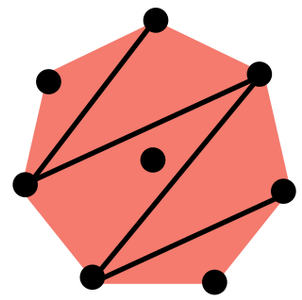


Theorem 2.4 (Carathéodory)

$conv(\mathcal{P}) =$ Union of all convex combinations with at most $(d+1)$ points in $\mathcal{P}$

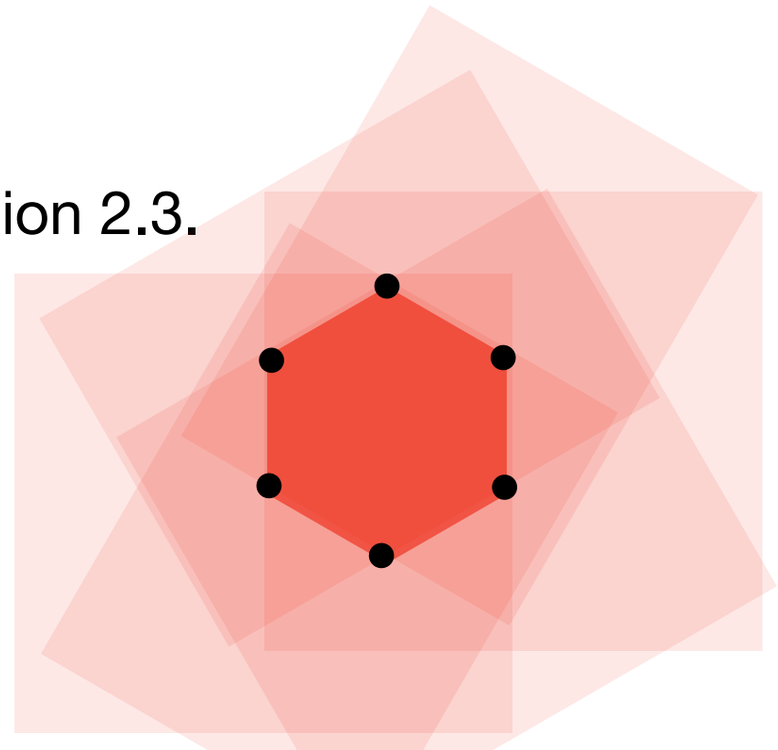
Corollary 2.5

$\mathcal{P} \subset \mathbb{R}^2 \Rightarrow conv(\mathcal{P})$ union of all $\triangle(p,q,r)$ with $p,q,r \in \mathcal{P}$.



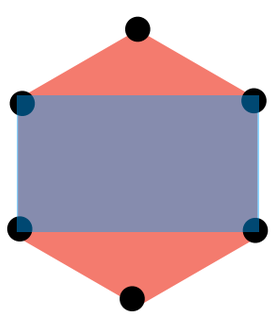
Lemma 2.6

The following definitions are equivalent to Definition 2.3.

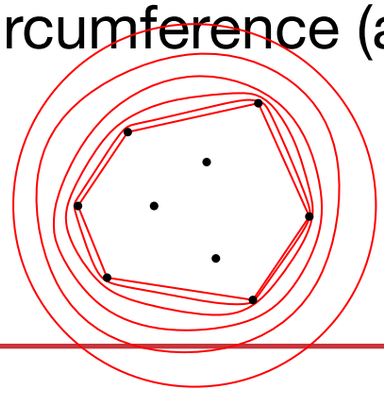


1. $conv(\mathcal{P}) := \bigcap_{P \supset \mathcal{P}, P \text{ convex}} P$

2. For $d = 2$: $\nexists$ convex polygon $P : \mathcal{P} \subseteq P \subsetneq \text{conv}(\mathcal{P})$



3. For $d = 2$: $conv(\mathcal{P}) :=$ conv. polygon P with minimal circumference (area) with $\mathcal{P} \subset P$

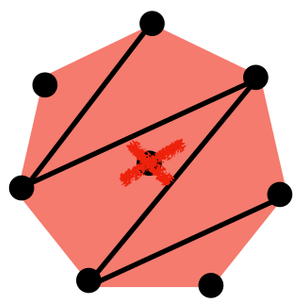


- From now on $\mathcal{P} \subset \mathbb{R}^2$
- First Approach: Find vertices of $conv(\mathcal{P})$ by elimination
- Negation of Corollary 2.5:

x not vertex of $conv(\mathcal{P})$

$\Updownarrow$

$\exists p_i, p_j, p_k \in \mathcal{P} : x$ = non-trivial convex combination of p_i, p_j, p_k.

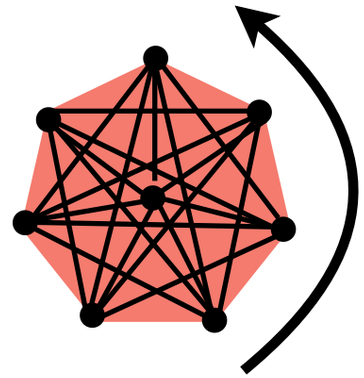


Algorithm 2.7

1: for (all triples (p_i, p_j, p_k) of points in $\mathcal{P}$) do
2: for (all points in $\in \mathcal{P}$) do
3: if (p lies in the inside of $\Delta(p_i, p_j, p_k)$
or on a boundary edge of $\Delta(p_i, p_j, p_k)$) then
4: mark p as an interior point.
5: $\mathcal{P}' := \{p \in \mathcal{P} \mid$ is unmarked $\}$;

Analysis of Algorithm 2.7:

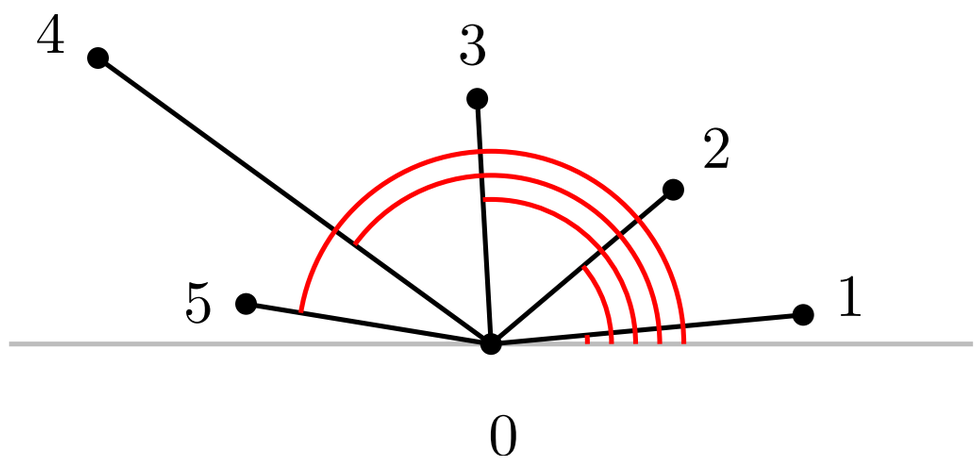
- $\binom{n}{3} \in \Theta(n^3)$ triangles



- Per triangle: $\Theta(n)$ further points
- Sort $\mathcal{O}(n)$ vertices
- Total runtime: $\mathcal{O}(n^4 + n \log n) = \mathcal{O}(n^4)$

Sorting criterion:

- CCW on $conv(\mathcal{P})$.
- Polar angle wrt y-minimal point in $\mathcal{P}$.
- Issue: Do we need trigonometry?



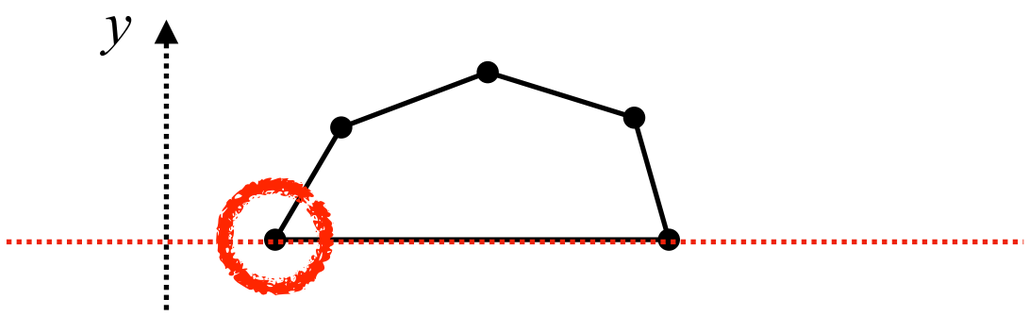
Lexicographic order

- Choose the „y-minimal“ point by lexicographic order:

$$(p.x, p.y) \leq_y (q.x, q.y) :\Leftrightarrow ((p.y < q.y) \vee ((p.y = q.y) \wedge (p.x \leq q.x)))$$

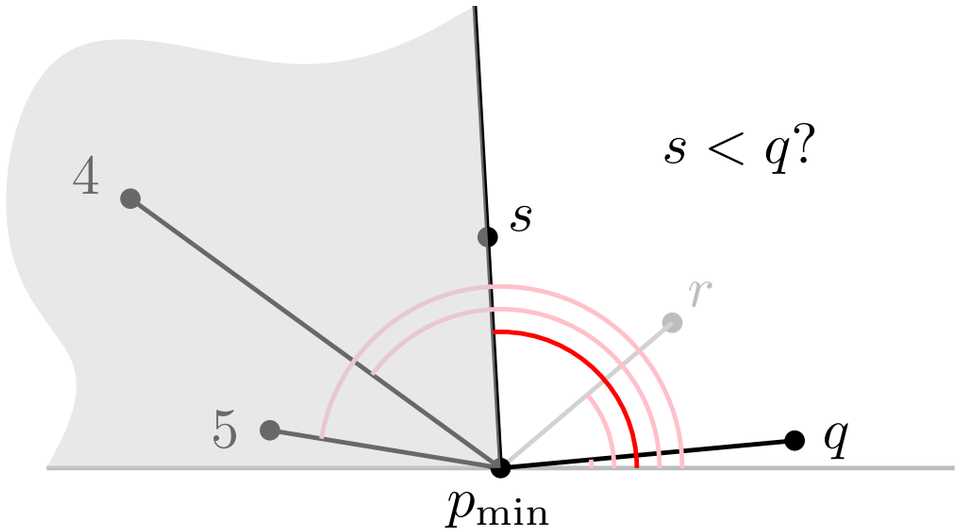
- Analogously: by x-coordinate

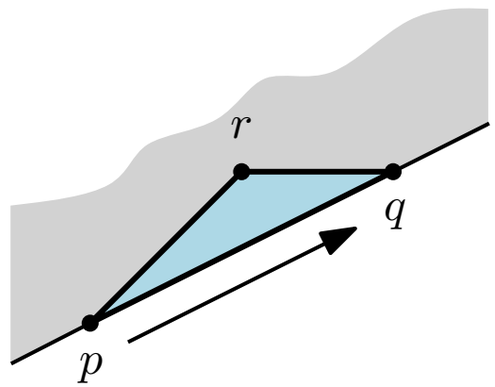
$$(p.x, p.y) \leq_x (q.x, q.y) :\Leftrightarrow ((p.x < q.x) \vee ((p.x = q.x) \wedge (p.y \leq q.y)))$$



Observation:

- Goal: CCW order
- Sort by polar angle.
- Sufficient: pairwise comparion of points s, q
- Check relative position of q wrt $\overline{p_{\min} s}$





Consequence:

- Predicate: $a \leq b :\Leftrightarrow ((a = p_{\min}) \vee (a = b) \vee (\textsc{LeftTurn}(p_{\min}, a, b) = \textsc{true}))$

1. Introduction and Definitions
2. Interlude: Algorithmic Paradigms
3. Jarvis' March
4. Quickhull
5. Divide-and-conquer and incremental construction
6. Graham's Scan
7. Optimal output-sensitive construction

Review: Sorting

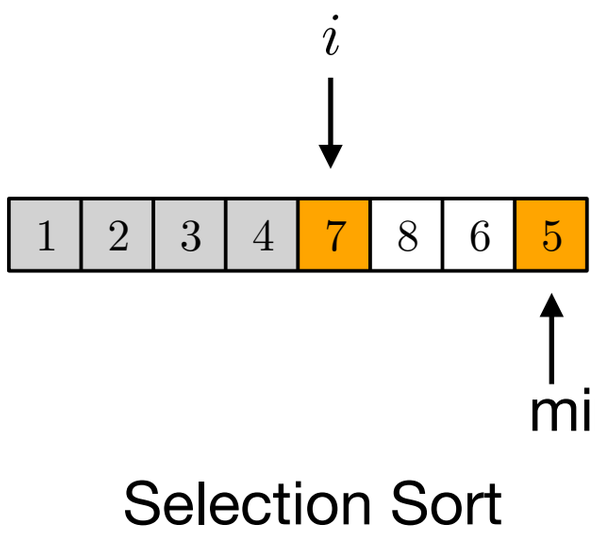
- Algorithms and Data Structures 1
- Various algorithmic paradigms

Sorting algorithms:

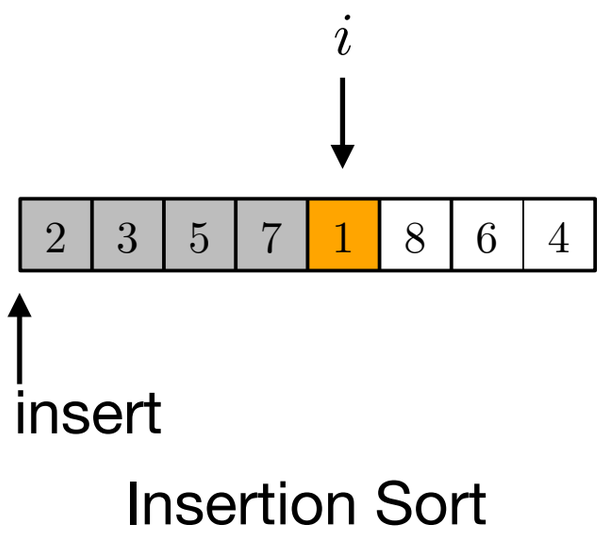
- Incremental methods:
  - Bubble Sort, Selection Sort, Insertion Sort
- Divide-and-conquer methods:
  - Quicksort, Mergesort
- Methods based on data structures:
  - Heapsort, sorting by AVL tree
- Other methods:
  - Bucket Sort, Shellsort, …

Difference: Selection Sort <-> Insertion Sort

- Selection Sort: Search in *unsorted* part of array
- Insertion Sort: Search in *sorted* part of array



Selection Sort

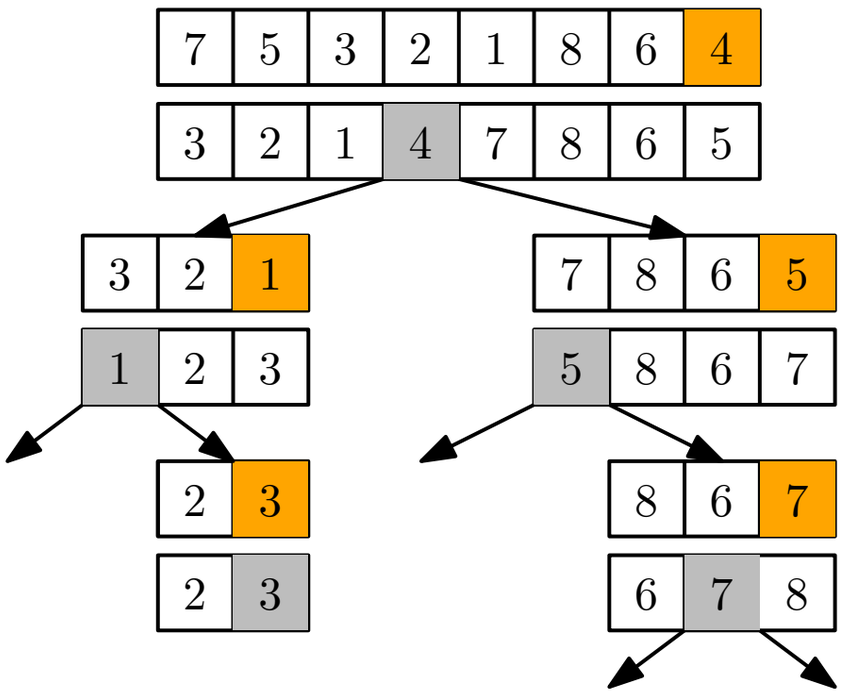


Insertion Sort

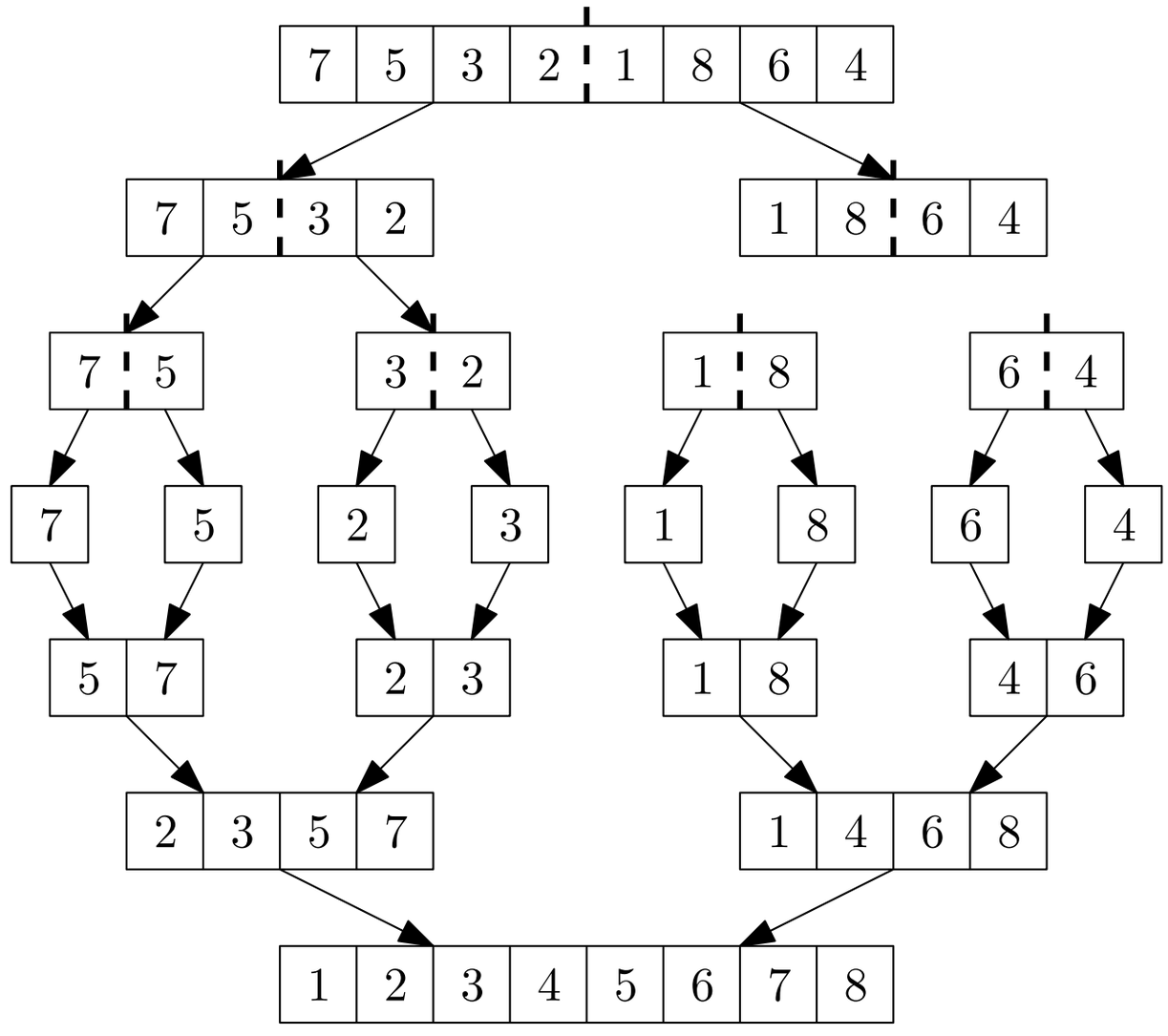
Approach:

- Selection Sort: Find next element for extending the order
- Insertion Sort: Insert next element, such that sequence remains sorted

- Split (Quicksort) or combine (Mergesort)

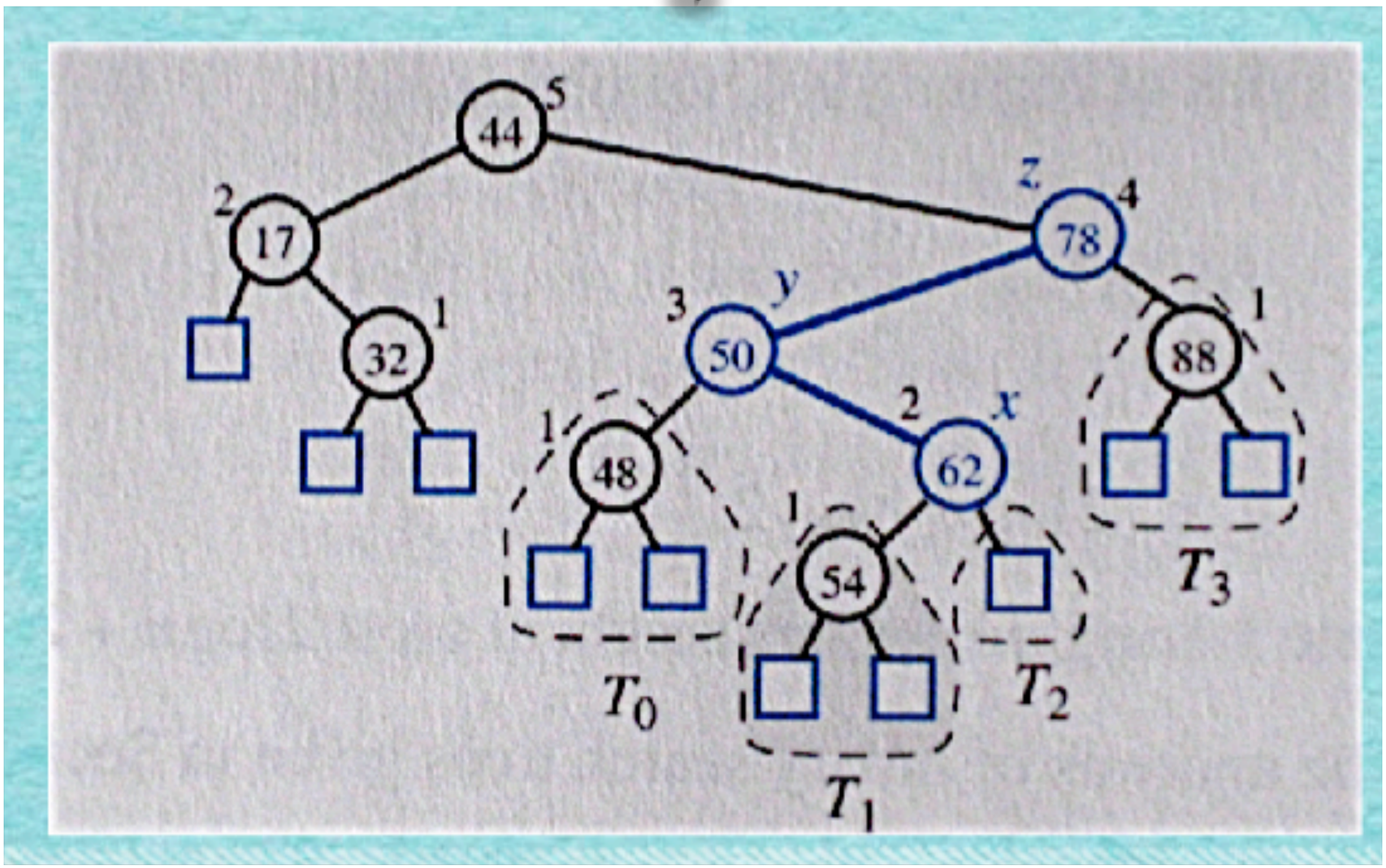


Quicksort



Mergesort

„Algorithms and Data Structures“: AVL-tree

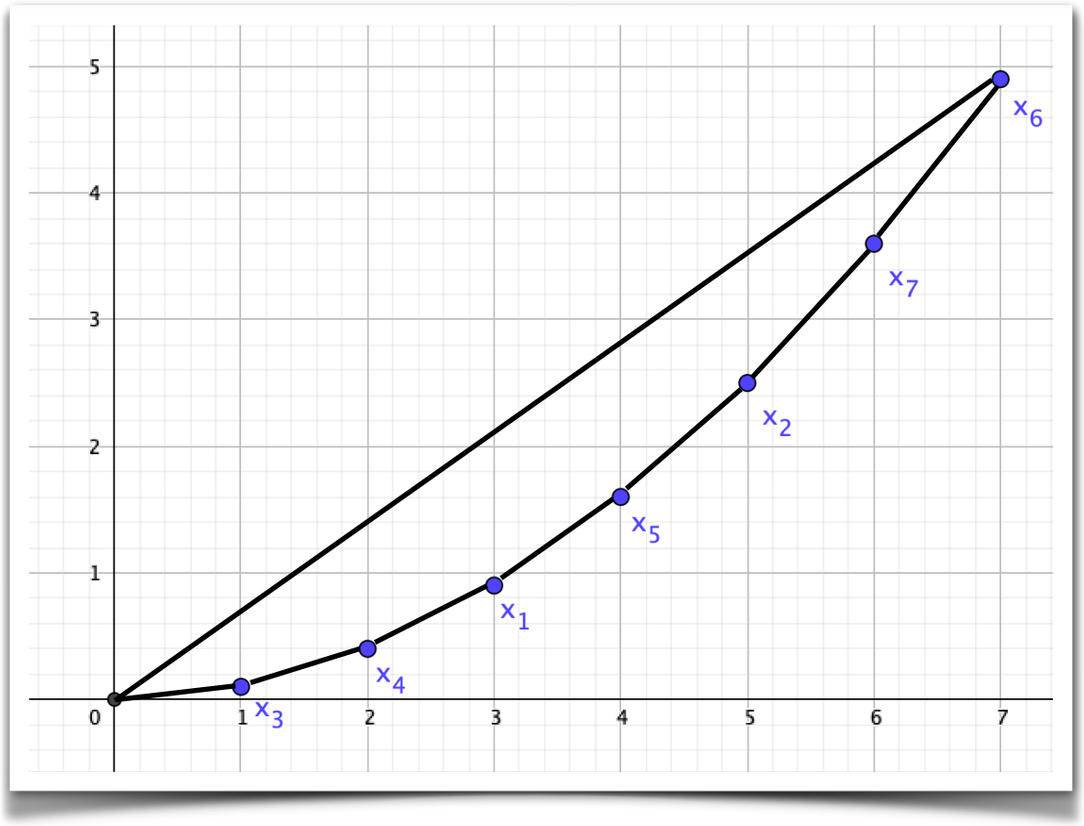


Theorem 1.18: Computing the convex hull takes $\Omega(n \log n)$ in certain models of computation.

Proof: Recall that comparison-based sorting takes $\Omega(n \log n)$.

Consider a set of n numbers, $x_1, \ldots, x_n$.

Map them to the points $(x_1, x_1^2) \ldots, (x_n, x_n^2)$.



3, 5, 1, 2, 4, 7, 6

The convex hull yields the sorted order of numbers.

1. Introduction and Definitions
2. Interlude: Algorithmic Paradigms
3. Jarvis' March
4. Quickhull
5. Divide-and-conquer and incremental construction
6. Graham's Scan
7. Optimal output-sensitive construction

INFORMATION PROCESSING LETTERS 2 (1973) 18--21. NORTH-HOLLAND PUBLISHING COMPANY

ON THE IDENTIFICATION OF THE CONVEX HULL OF A FINITE SET OF POINTS IN THE PLANE

R.A. JARVIS

The Australian National University, Department of Statistics, Box 4, Canberra, A.C.T. 2600, Australia

Received 6 December 1972

convex hull algorithm

1. Introduction

This paper presents an extremely simple algorithm for identifying the convex hull of a finite set of points in the plane in essentially, at most $n(m+1)$ operations for n points in the set and $m \leqslant n$ points on the convex hull. In most cases far less than $n(m+1)$ operations are necessary because of a powerful point deletion mechanism that can easily be included. The operations are themselves trivial (computationally inexpensive) and consist of angle comparisons only. Even these angle comparisons need not be actually carried out if an improvement suggested in a later section is implemented. Although Graham's algorithm [1] requires no more than $(n \log n)/\log 2 + Cn$ operations*, the operations are themselves more complex than those of the method presented here; in particular, Graham's method would not be as efficient for low m.

2. Geometric interpretation

The underlying method of the algorithm can be described simply: find an origin point outside the point set and swing a radius arm in an arbitrary direction until a point of the set is met; this point becomes

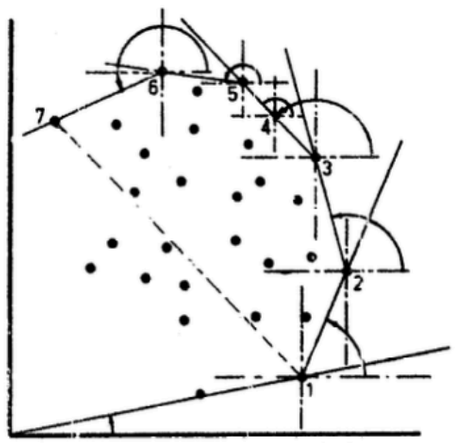


Fig. 1. Geometric interpretation of the algorithm.

the first point on the hull. Make this the new origin point and swing a radius arm from this point in the same direction as before till the next hull point is found. Repeat until the points are enclosed by the convex hull. Delete points from further consideration if

(i) they have already been identified as being on the convex hull,

(ii) they lie in the area enclosed by a line from the first to the last convex hull point found and the lines joining the convex hull points in the sequence found.

Fig. 1 illustrates this geometric interpretation.

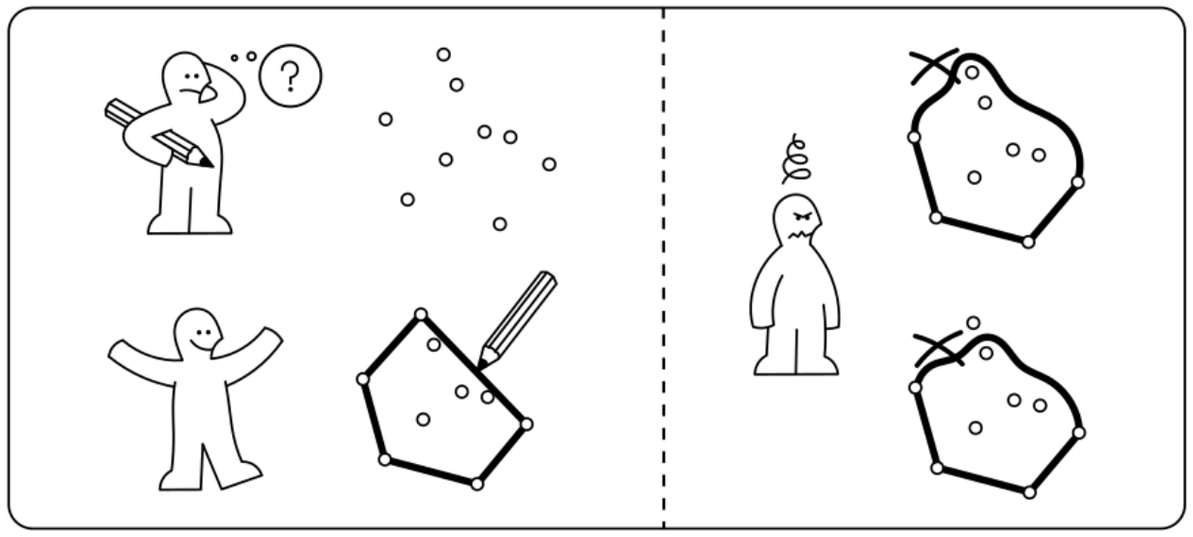
* To quote Graham, "C is a small positive constant which depends on what is meant by an 'operation'". In fact, C is distributed over the five basic steps of Graham's algorithm and his paper should be consulted for detailed interpretation.

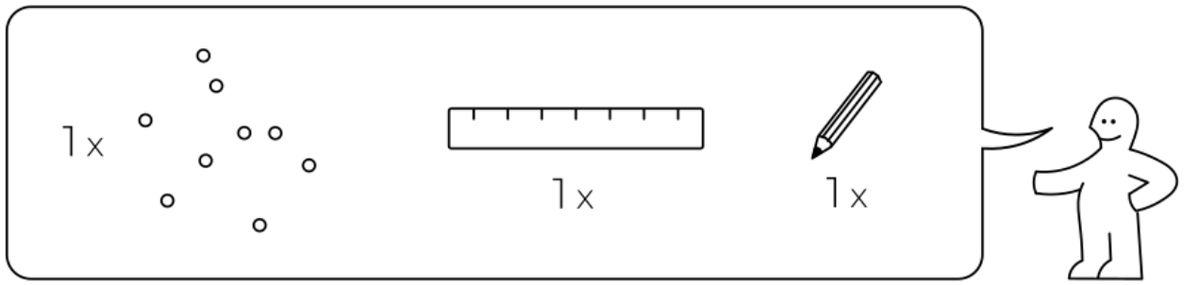

IDEA
A series of nonverbal
algorithm assembly instructions.

GIFT WRÄPPING

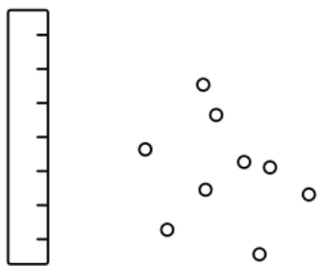
idea-instructions.com/gift-wrapping/
Based on a guest contribution by Christoph Hansknecht – v1.1, CC by-nc-sa 4.0

IDEA

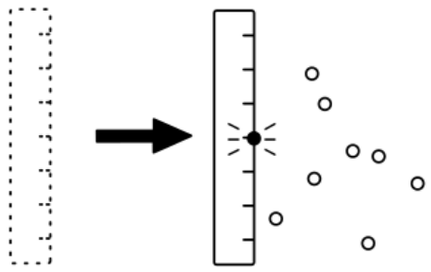




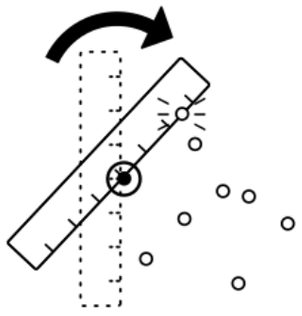
1a



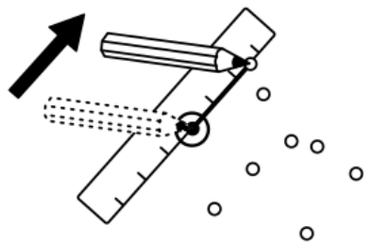
1b



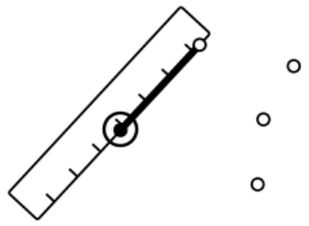
1c



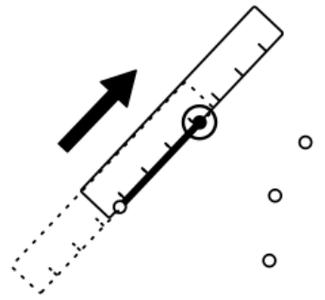
1d



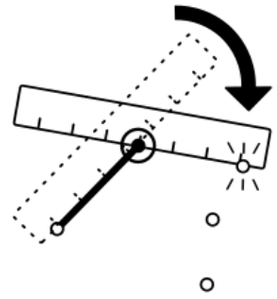
2a



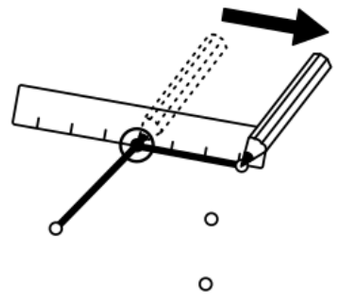
2b



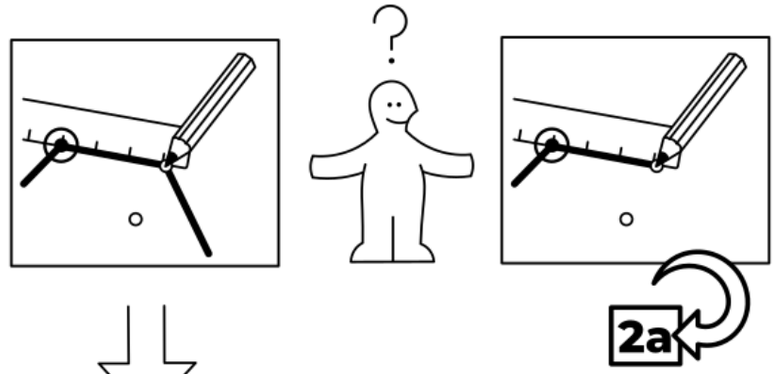
2c



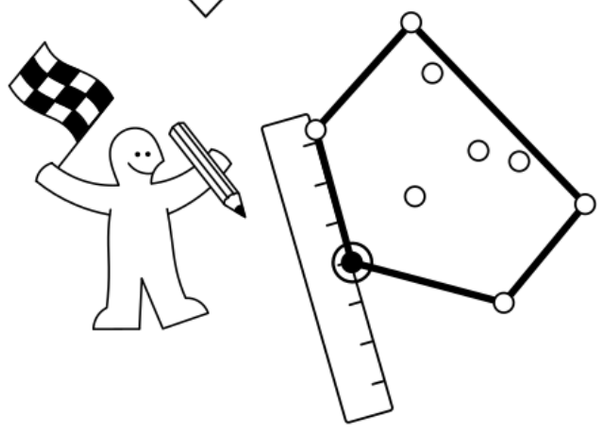
2d



3



4



Thank you!