

Fine Grained Analysis: examples and challenges

Jérémy Barbay

Departamento de Ciencias de la Computacion
Universidad de Chile

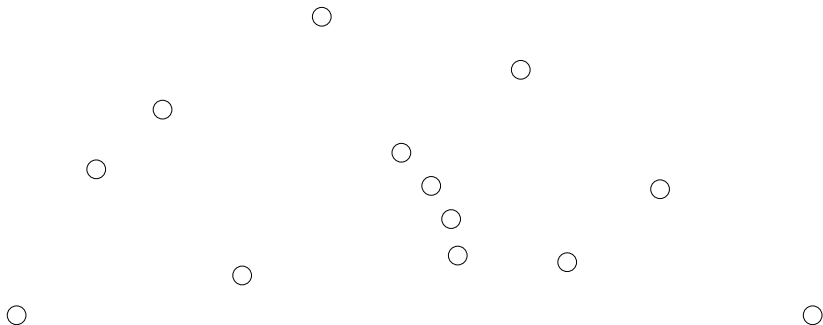
Outline

- 1 The Convex Hull Paradox
 - $O(n \lg n)$
 - $O(nh)$
 - Worst Case Complexity?
- 2 Fine analysis of the convex hull
 - $O(n \lg h)$ in 2D
 - $O(n \lg h)$ in 3D
 - $O(nH(C))$, instance optimal
- 3 Similar Paradoxes

Outline

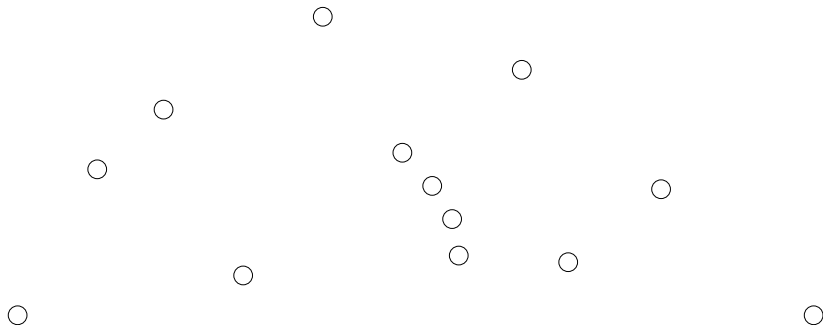
- 1 The Convex Hull Paradox
 - $O(n \lg n)$
 - $O(nh)$
 - Worst Case Complexity?
- 2 Fine analysis of the convex hull
 - $O(n \lg h)$ in 2D
 - $O(n \lg h)$ in 3D
 - $O(nH(C))$, instance optimal
- 3 Similar Paradoxes

The Planar Convex Hull



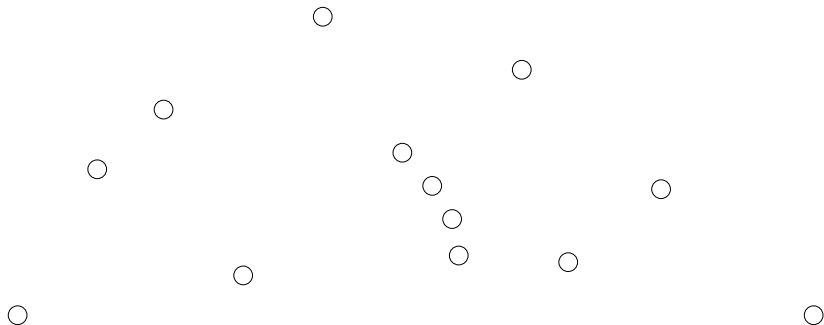
- Can one of you define it?
- What is the best complexity known for it?

2d Convex Hull in $O(n \lg n)$



- 1 Sort the points by x -coordinates;
- 2 Scan them, backtracking if necessary.

2d Convex Hull in $O(nh)$



- 1 Find the left-most point
- 2 Compute the $n - 1$ slopes with the other points
- 3 Choose the highest slope
- 4 Iterate

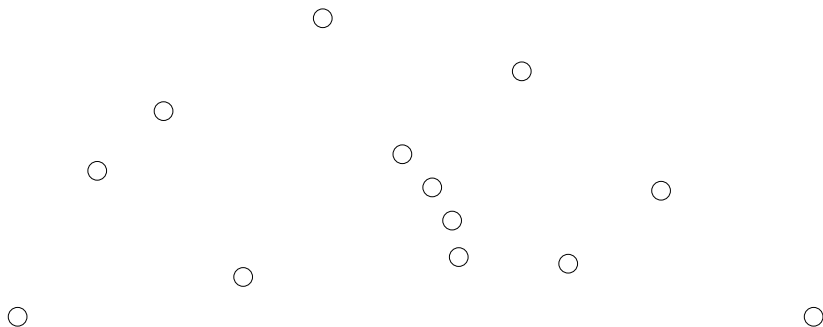
Worst case Complexity of 2d Convex Hull

- Question ill-defined: the worst case over what?
 - all instances of fixed size n ?
 - all instances of fixed input size n and output size h ?
- For each we have distinct lower bounds:
 - $\Omega(n \lg n)$, which is tight; and
 - $\Omega(n \lg h)$, which is **not** tight!
- So what is the complexity of 2d convex hull?

Outline

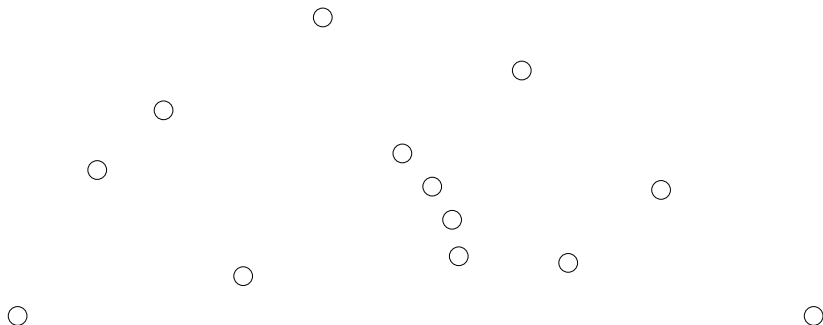
- 1 The Convex Hull Paradox
 - $O(n \lg n)$
 - $O(nh)$
 - Worst Case Complexity?
- 2 Fine analysis of the convex hull
 - $O(n \lg h)$ in 2D
 - $O(n \lg h)$ in 3D
 - $O(nH(C))$, instance optimal
- 3 Similar Paradoxes

2d Convex Hull in $O(n \lg h)$ in 2D



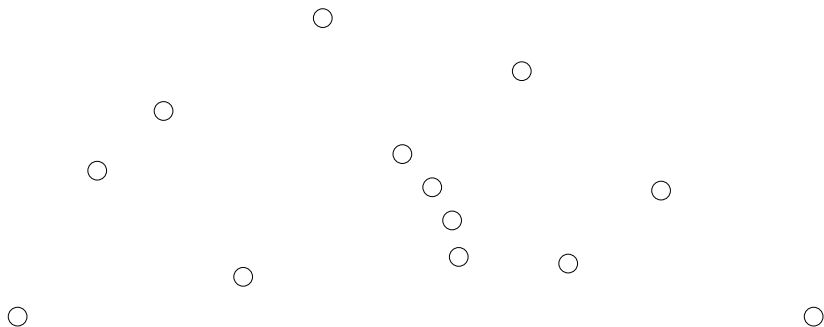
- 1 Compute the point m of median x -coordinate;
- 2 Partition the points by $m.x$;
- 3 Compute the highest edge (a, b) intersecting the line $x = m.x$;
- 4 Recurse on each side;

2d Convex Hull in $O(n \lg h)$ in 3D



- 1 Start with a small guess for h ;
- 2 Group the instances in n/h x -sorted groups of size h ;
- 3 Simulate the $O(nh)$ algorithms on the groups;
- 4 If it did not suffice, merge the group two by two and iterate.

Convex Hull in $O(n(1 + H(C)))$



- Algorithm: a variant of [Kirkpatrick, Seidel]
 - 1 Compute the points leftmost l and rightmost r ;
 - 2 Compute the point m of median x -coordinate;
 - 3 Compute the highest edge (a, b) intersecting the line $x = m$;
 - 4 Remove all points contained in the polygon (l, a, b, r) ;
 - 5 Recurse on each side;

Instance Optimality: definitions

Definition (Instance Optimality)

An algorithm is **instance-optimal** if its cost is at most a constant factor from the cost of any other algorithm A' running on the same input, for every input instance.

Unfortunately, for many problems, this requirement is too stringent.

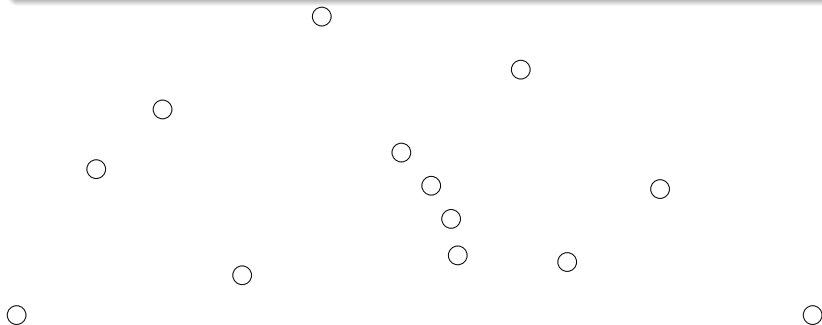
Definition (Input Order Oblivious Instance Optimality)

For a set S of n elements in \mathcal{D} , let $T_A(S)$ denote the maximum running time of A on input σ over all $n!$ possible permutations σ of S . Let $\text{OPT}(S)$ denote the minimum of $T_{A'}(S)$ over all correct algorithms $A' \in \mathcal{A}$. If $A \in \mathcal{A}$ is a correct algorithm such that $T_A(S) \leq O(1) \cdot \text{OPT}(S)$ for every set S , then we say A is **instance-optimal in the order-oblivious setting**.

Certificate and Instance Optimal Proof

Definition (Certificate)

A *Certificate* for an instance I and a solution S is the description of a sequence of steps to **check** the validity of S for I .



Example

For the convex hull, a list of triangles and the points they cover.

Outline

- 1 The Convex Hull Paradox
 - $O(n \lg n)$
 - $O(nh)$
 - Worst Case Complexity?
- 2 Fine analysis of the convex hull
 - $O(n \lg h)$ in 2D
 - $O(n \lg h)$ in 3D
 - $O(nH(C))$, instance optimal
- 3 Similar Paradoxes

Optimal Prefix Free Codes [In Progress]

- $O(n \lg n)$ classical algorithm.
- $O(n)$ algorithm when frequencies are sorted.
- $O(n)$ algorithm when frequencies are all within a factor of 2.
- $O(n)$ algorithm when frequencies are all distinct by factor of 2.
- Adaptive Results for k distinct code lengths:
 - Belal and Elmasry claim $O(nk)$ in STACS 2006.
 - Belal and Elmasry claim $O(n4^k)$ in ARXIV 2012.
- A lower bound of $\Omega(n \lg k)$ in the worst case over instances resulting in k distinct code lengths.
- Conjectures:
 - $O(n \lg k)$ adaptive algorithm?
 - $O(nH(n_1, \dots, n_k))$ instance optimal algorithm?
 - $O(n)$ algorithm in word-RAM?

Optimal Minimax Trees [Open]

- Tree minimizing the max weight+height of a leaf.
- $O(n \lg n)$ classical algorithm [Golumbic];
- $O(n)$ algorithm when weights partially sorted by fractional part [Drmota, Szpankowski];
- $O(nd \lg \lg n)$ where d is the number of distinct values $\lceil w_i \rceil$ [Kirkpatrick and Klawe]
- $O(n)$ algorithm in word-RAM [Gawrichowski, Gagie]!

Optimal Alphabetic Binary Search Tree [Open]

- $O(n \lg n)$ classical Hu-Tucker algorithm;
- $o(n \lg n)$ algorithms in many particular cases;
- $O(n)$ algorithm when frequencies “can be sorted in linear time”;
- A lower bound of $\Omega(n \lg k)$ in the worst case over instances resulting in k distinct code lengths.

Summary

- $O(nk)$ and $O(n \lg n)$ suggests $O(n \lg k)$
- and (Input Order Oblivious) **Instance Optimality**.
- Outlook
 - Input Order Adaptive Instance Optimality.
 - Full Instance Optimality (Kolmogorov's complexity?)
 - 1-instance optimality!