

# From Fine Grained Analysis to Instance Optimality

Jérémy Barbay

jeremy@barbay.cl

[2016-06-21 Tue]@Ljubljana

*Informal and Interactive!*

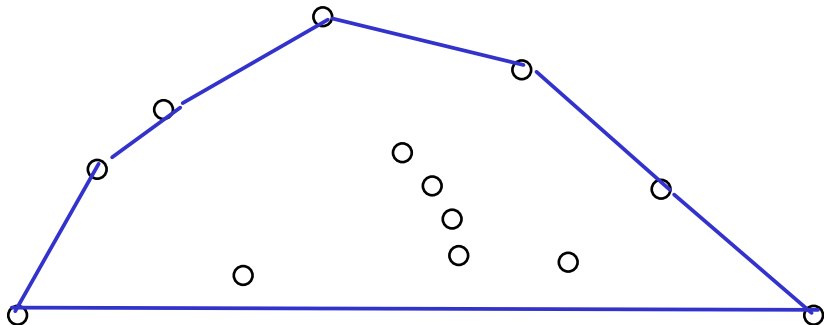
# Outline

- 1 The Convex Hull Paradox
  - $O(n \lg n)$
  - $O(nh)$
  - Worst Case Complexity?
- 2 Fine grained 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
  - Prefix Free Codes and variants
  - Planar Graph Algorithms

# Outline

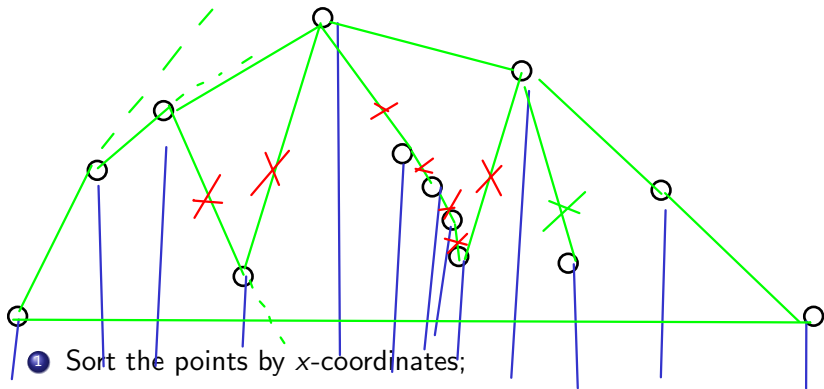
- 1 The Convex Hull Paradox
  - $O(n \lg n)$
  - $O(nh)$
  - Worst Case Complexity?
- 2 Fine grained analysis of the convex hull
  - $O(n \lg h)$  in 2D
  - $O(n \lg h)$  in 3D
  - $O(nH(C))$ , instance optimal
- 3 Similar Paradoxes
  - Prefix Free Codes and variants
  - Planar Graph Algorithms

# The Planar Convex Hull



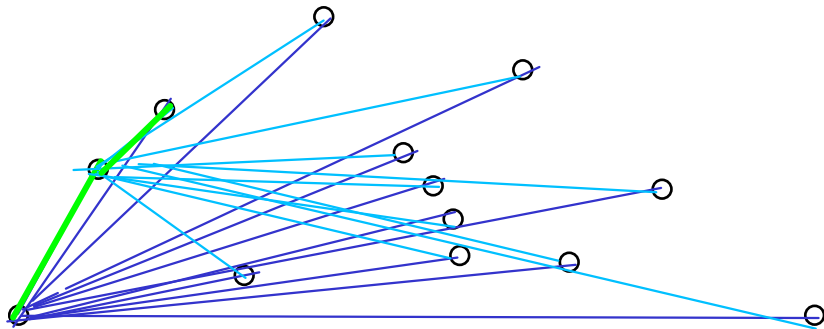
- Can one of you define it?
- What is the best complexity known for it?

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



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

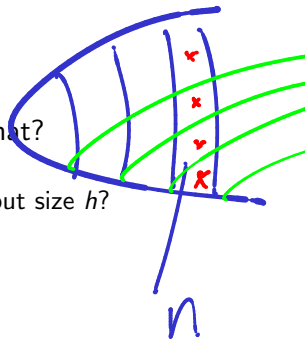
## 2d Convex Hull in $O(nh)$



- 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?

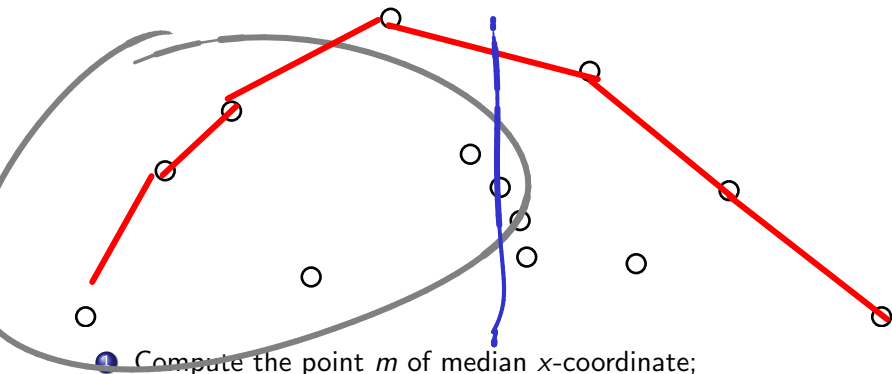


# Outline

- 1 The Convex Hull Paradox
  - $O(n \lg n)$
  - $O(nh)$
  - Worst Case Complexity?
- 2 Fine grained 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
  - Prefix Free Codes and variants
  - Planar Graph Algorithms

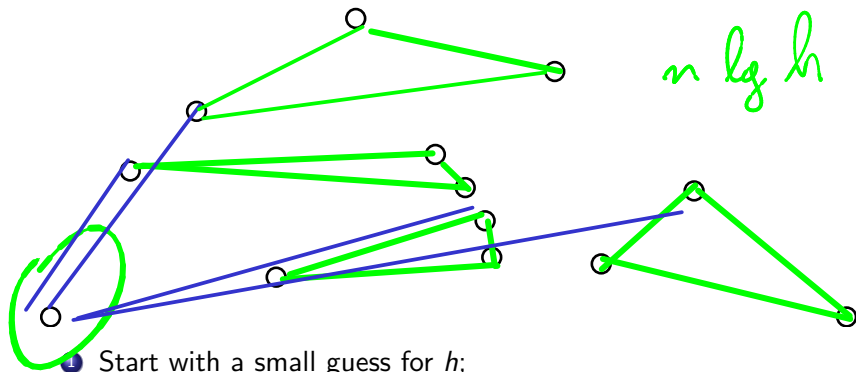


## 2d Convex Hull in $O(n \lg h)$ in 2D



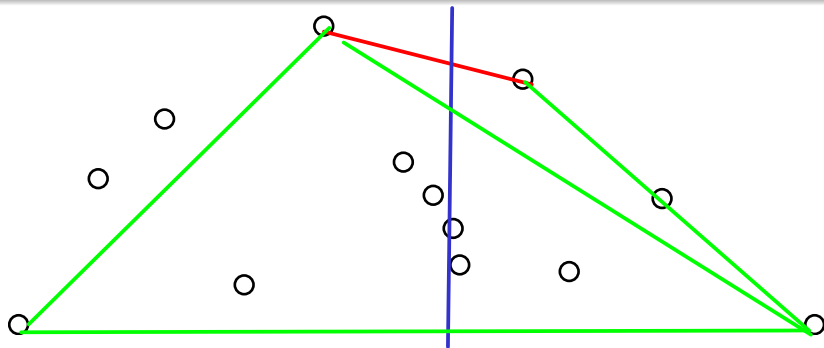
- 1 Compute the point  $m$  of median  $x$ -coordinate;
- 2 Partition the points by  $m.x$ ;
- 3 Compute the highest edge  $(a, b)$  intersecting the line  $x = m.x$ ;
- 4 Recurse on each side;

## 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  $x = m.x$ ;
  - 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.

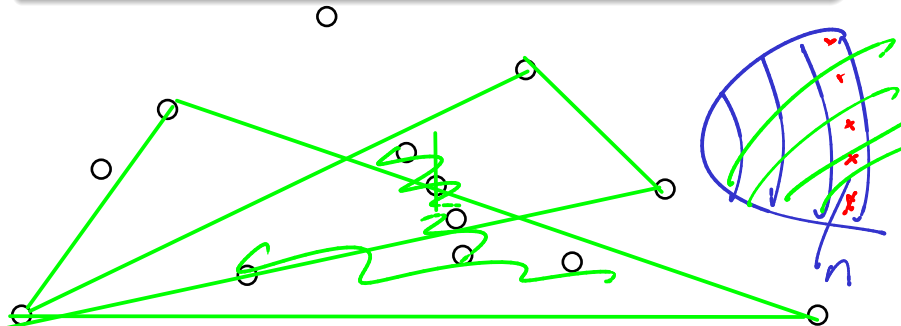
### Definition (Input Order Oblivious Instance Optimality)

For a set  $S$  of  $n$  elements in  $\mathcal{D}$ , let  $T_A(S)$  denote the maximum running time of  $A$  on input  $\sigma$  over all  $n!$  possible permutations  $\sigma$  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 grained 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
  - Prefix Free Codes and variants
  - Planar Graph Algorithms

## Similar Paradoxes

- PREFIX FREE CODES (aka Huffman) [Thursday's talk!]
- MINIMAX TREES [In Progress]
- ALPHABETIC BINARY SEARCH TREES (aka Hu Tucker) [Open]
- DIRECTED MAX  $(s, t)$  FLOW [Open]
- UNDIRECTED MIN  $(s, t)$  CUT [Open]
- (...)

# 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 (vs  $O(n \lg \lg n)$  for int. sorting)?



# Optimal Minimax Trees [Open]

- Classical:
  - Tree minimizing the max weight+height of a leaf.
  - $O(n \lg n)$  classical algorithm [Golumbic];
- Fine Grained Analysis Results:
  - $O(n)$  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;
- Easy Cases:
  - $o(n \lg n)$  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.
- Conjectures:
  - $O(n \lg k)$  adaptive algorithm?
  - $O(nH(n_1, \dots, n_k))$  instance optimal algorithm?
  - **$O(n)$  algorithm in word-RAM?**

# Directed Max $(s, t)$ Flow [Open]

- $O(n \lg n)$  classical algorithm;
- $O(n)$  well known algorithm when  $s$  and  $t$  share a face;
- $O(nk)$  new algorithm, where  $k$  is the number of edges between  $s$  and  $t$ ;

# Undirected Min $(s, t)$ Cut [Open]

- $O(n \lg n)$  well known algorithm
- $O(n \lg k)$  recent algorithm (STACS 2011!)
- Can this be improved?

# Summary

- $O(nk)$  and  $O(n \lg n)$  suggests  $O(n \lg k)$  and
- (Input Order Oblivious) **Instance Optimality**.
- Incoming big survey (2017).
- Outlook
  - Synergistic Algorithms
  - Input Order Adaptive Instance Optimality, and Full Instance Optimality.
  - 1-instance optimality.

# Bibliography



Afshani, P., Barbay, J., and Chan, T. M. (2009).

Instance-optimal geometric algorithms.

In *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 129–138. IEEE Computer Society.



Barbay, J. (2013).

From time to space: Fast algorithms that yield small and fast data structures.

In Brodnik, A., López-Ortiz, A., Raman, V., and Viola, A., editors, *Space-Efficient Data Structures, Streams, and Algorithms (IaNFest)*, volume 8066 of *Lecture Notes in Computer Science*, pages 97–111. Springer.



Barbay, J., Chan, T. M., Navarro, G., and Pérez-Lantero, P. (2013).

Maximum-weight planar boxes in  $o(n^2)$  time (and better).

In *CCCG*. Carleton University, Ottawa, Canada.



Barbay, J. and Pérez-Lantero, P. (2015).

Adaptive computation of the Swap-Insert Correction Distance.

In *Proceedings of the Annual Symposium on String Processing and Information Retrieval (SPIRE)*.