CSE520: Computational Geometry I
Lecture 4
Topological Lower Bounds I

Antoine Vigneron

Ulsan National Institute of Science and Technology

June 15, 2020

Introduction

- References:
  - Textbook by Preparata and Shamos.
  - Dave Mount's lecture notes, Lecture 26.
  - Ben-Or's paper.

Introduction

In the algorithms course (CSE331), you learned that:

Theorem (Lower bound for sorting)

Any comparison-based sorting algorithm makes $\Omega(n \log n)$ *comparisons in the worst case.*

Proof (sketch):

- Model the algorithm as a binary decision tree.
- Each internal node is a comparison, branching to its two children.
- There are $n!$ possible outcomes, i.e. permutation of the input.
- Hence there are at least $n!$ leaves.
- So the tree has height $\Omega(\log(n!)) = \Omega(n \log n)$.

Introduction

In Lecture 2, we showed that it yields the same $\Omega(n \log n)$ lower bound for computing a convex hull.

- Proof: After mapping a set of numbers to a parabola, the numbers appear in sorted order along the convex hull.
- So the argument holds because a convex hull algorithm outputs a point sequence.

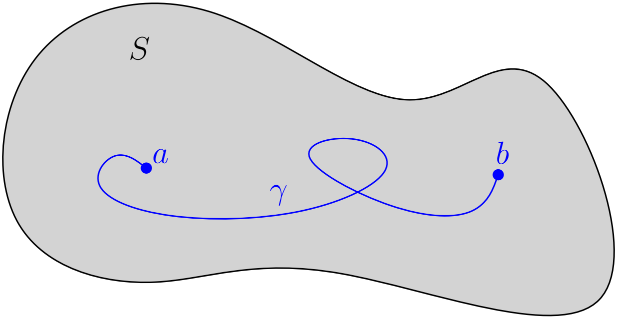
This argument does not work if the output has constant size.

Examples

- Intersection detection. (Output: a Boolean.)
- Diameter. (Output: a real number.)

We will give a different technique that yields an $\Omega(n \log n)$ lower bound for these two problems, and others.

Connectedness



- Let S be a subset of $\mathbb{R}^n$. A *path* γ in S is a continuous function $\gamma : [0, 1] \to S$.
- We say that it is a path from $a = \gamma(0)$ to $b = \gamma(1)$.
- The set S is *connected* if there is a path between any two points in S.

Connectedness

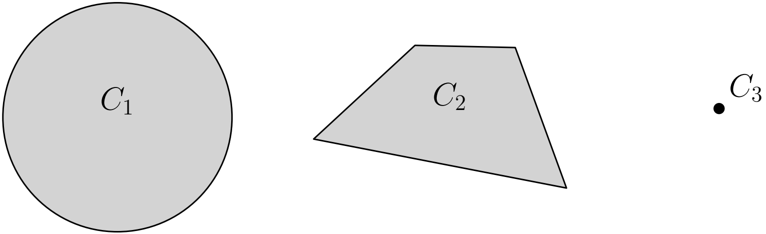
Example

Any convex set is connected.

- Why?
- By definition, any two points are connected by a straight-line path.

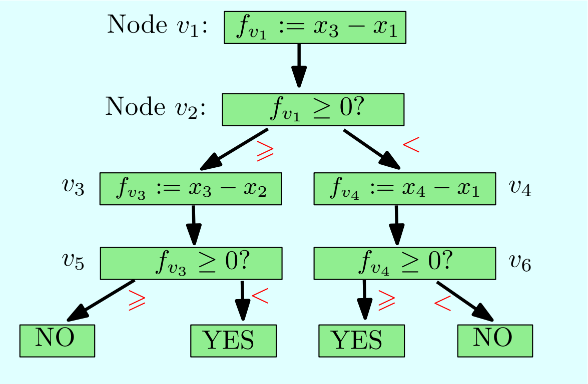
Connected Components

- A *connected component* of S is a maximal connected subset of S.
- The connected components of S form a partition of S.



- The set above has three connected components C_1, C_2 and C_3.

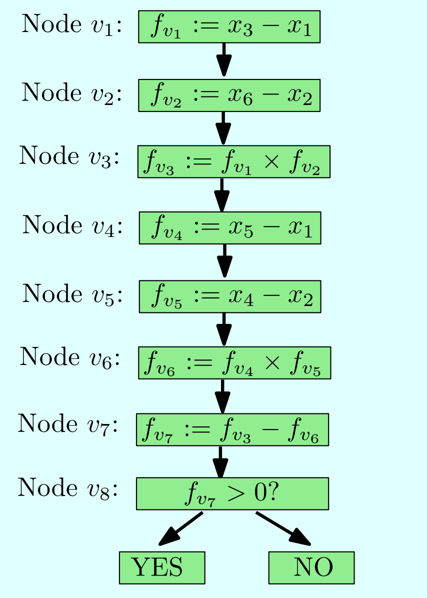
Algebraic Computation Trees



Example (1)

Given $(x_1, x_2, x_3, x_4) \in \mathbb{R}^4$ such that $x_1 \leqslant x_2$ and $x_3 \leqslant x_4$, this Algebraic Computation Tree (ACT) decides whether $[x_1, x_2] \cap [x_3, x_4] \neq \emptyset$.

Algebraic Computation Trees



Example (CCW predicate)

Given $(x_1, x_2, x_3, x_4, x_5, x_6) \in \mathbb{R}^6$, this ACT decides whether the triangle $((x_1, x_2), (x_3, x_4), (x_5, x_6))$ is counterclockwise.

Algebraic Computation Trees

Formal definition from Ben-Or's paper:

Definition (Algebraic computation tree)

An algebraic computation tree with input $(x_1, \ldots, x_n) \in \mathbb{R}^n$ is a binary tree T with a function that assigns:

- to any vertex v with exactly one child an operational instruction of the form
$$f_v := f_v^1 \circ f_v^2 \text{ or } f_v := c \circ f_v^1 \text{ or } f_v := \sqrt{f_v^1}$$
where $f_v^i = f_{v_i}$ for an ancestor v_i of v, or $f_v^i \in \{x_1, \ldots, x_n\}$, $\circ \in \{+, -, \times, /\}$, and $c \in \mathbb{R}$ is a constant.
- to any vertex v with two children (branching vertex) a test instruction of the form
$$f_v^1 > 0 \text{ or } f_v^1 \geqslant 0 \text{ or } f_v^1 = 0.$$
where f_v^1 is f_{v_1} for an ancestor v_1 of v, or $f_v^1 \in \{x_1, \ldots, x_n\}$.
- to any leaf an output YES or NO.

Algebraic Computation Trees

Informally:

- Given an input $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, the program traverses a path $P(x)$ from the root to a leaf of T.
- Along the path, it applies operations $+, -, /, \times, \sqrt{\cdot}$ to input numbers x_i or intermediate results obtained at previous nodes along $P(x)$.
- It may also branch using a test $>, \geqslant, =$.
- At the leaf, it outputs YES or NO.

Definition

Let T be an algebraic computation tree with input $x \in \mathbb{R}^n$. Let $W \subset \mathbb{R}^n$ be the set of points $x \in \mathbb{R}^n$ such that T outputs YES. We say that T *decides* W.

Topological Lower Bound

Theorem (Ben-Or, 1983)

Any algebraic computation tree that decides a set $W \subset \mathbb{R}^n$ has height $\Omega(\log(\#W) - n)$, where $\#W$ is the number of connected components of W.

Interpretation:

- The height of an ACT is its worst-case running time.
- A program can often be *unfolded* onto an ACT.
  - Then its worst-case running time is at least the height of the ACT.
- But some operations cannot be simulated by an ACT in $O(1)$ time. Examples:
  - The floor function.
  - Bitwise operations on integers (AND, OR, XOR).
  - Random number generation.

Element Distinctness

Problem (Element Distinctness)

Determine whether the elements of a list of numbers are distinct. That is, given $(x_1, \ldots, x_n) \in \mathbb{R}^n$, determine whether $x_i \neq x_j$ for all $i \neq j$.

- Let W^+ denote the set of positive instances of Element Distinctness:

$$W^+ = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_i \neq x_j \text{ for all } i \neq j\}.$$

- Note that an instance of Element Distinctness is modeled as a *single point* in $\mathbb{R}^n$. Hence W^+ is a subset of $\mathbb{R}^n$.
- How many connected components are there in W^+?

Element Distinctness

Lemma

The set W^+ of positive instances of Element Distinctness has exactly $n!$ connected components.

- We now prove this lemma.
- A *permutation* of $\{1, 2, \dots n\}$ is a bijection $\sigma : \{1, 2, \dots, n\} \to \{1, 2, \dots, n\}$.
- We denote $\sigma_i = \sigma(i)$.

Example

When $n = 2$, there are two permutations: The identity I such that $I_1 = 1$ and $I_2 = 2$, and the permutation α such that $\alpha_1 = 2$ and $\alpha_2 = 1$.

Element Distinctness

- For any permutation σ of $\{1, \ldots, n\}$, we consider the set

$$W_\sigma = \{(x_1, \ldots, x_n) \mid x_{\sigma_1} < x_{\sigma_2} < \cdots < x_{\sigma_n}\}.$$

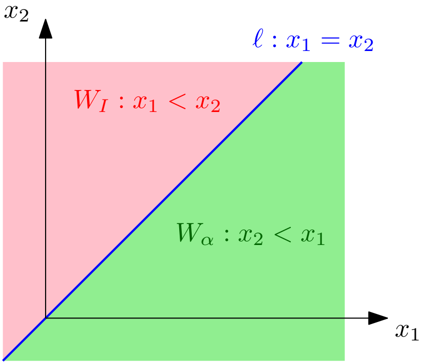
Example

When $n = 2$, there are two such sets.

- $W_I = \{(x_1, x_2) \mid x_1 < x_2\}$
- $W_\alpha = \{(x_1, x_2) \mid x_2 < x_1\}$

- So W_σ is the set of points with distinct coordinates, and such that the order of these coordinates is given by σ.
- We will argue that these sets are the connected components of W^+.

Element Distinctness



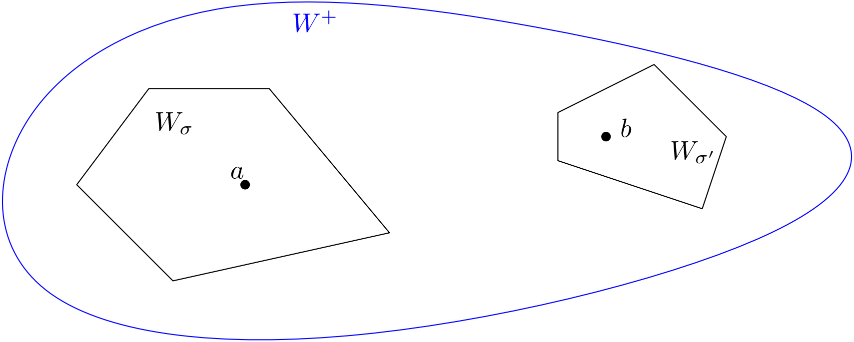
Example

When $n = 2$, the two connected components of $W^+ = \mathbb{R}^2 \setminus \ell$ are the two open halfplanes W_I and W_α.

Element Distinctness

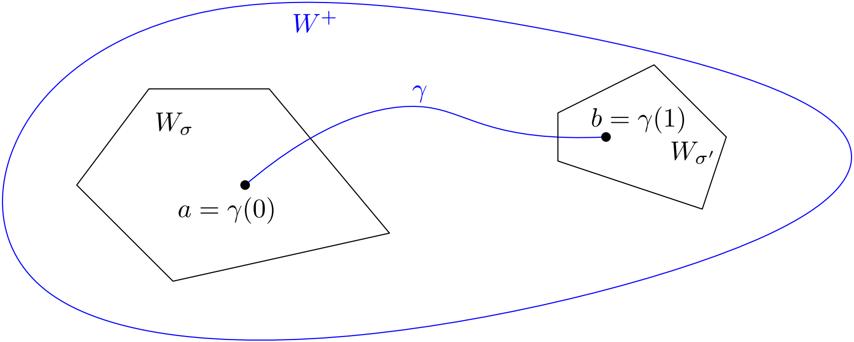
- The sets W_σ form a partition of W^+. Proof:
  - Each set W_σ is nonempty because the point x such that $x_{\sigma_i} = i$ for all i is in W_σ.
  - When $\sigma \neq \sigma'$, we have $W_\sigma \cap W_{\sigma'} = \emptyset$ because σ and σ' give different orders for the coordinates of the points.
  - Any $x \in W^+$ belongs to a set W_σ, because there must be a permutation σ such that $x_{\sigma_1} < \cdots < x_{\sigma_n}$. (Just sort these numbers.)
- The set W_σ is connected because it is convex. It is convex because it is the intersection of the halfspaces $x_{\sigma_i} < x_{\sigma_{i+1}}$, which are convex.
- In order to prove that W_σ is a connected component of W^+, we still need to prove that it is maximal.

Element Distinctness



- For sake of contradiction, suppose W_σ is not maximal.
- So there exists a connected set W^* such that $W_\sigma \subsetneq W^* \subseteq W^+$.
- Let $a \in W^+$ and $b \in W^* \setminus W_\sigma$.
- So $b \in W_{\sigma'}$ for some $\sigma' \neq \sigma$.

Element Distinctness



- So there is a path from a to b within W^+.
- There exists i, j such that $a_i < a_j$ and $b_i > b_j$.
- Let $g(t) = \gamma_i(t) - \gamma_j(t)$.
- We know that $g(0) < 0$ and $g(1) > 0$.

Element Distinctness

- As g is continuous, there exists $s \in [0,1]$ such that $g(s) = 0$.
- But then $\gamma(s) \notin W^+$ because $x_i(\gamma(s)) - x_j(\gamma(s)) = 0$.
- It contradicts the fact that γ is in W^+.

- So we have prove that the sets W_σ are the connected components of W^+.
- As there are $n!$ permutations of $\{1, 2, \ldots, n\}$, it means that W^+ has $n!$ connected components.

Element Distinctness

Theorem

In the algebraic computation tree model, the complexity of element distinctness is $\Theta(n \log n)$*.*

Proof.

By Ben-Or's theorem, since W^+ has $n!$ connected components, any ACT solving the element distinctness problem has height $\Omega(\log(n!) - n)$, which is $\Omega(n \log n)$.

Conversely, there is an ACT with depth $O(n \log n)$ that solves element distinctness: First sort the input, then check whether any two consecutive numbers are equal. For instance, mergesort can be unfolded into a $\Theta(n \log n)$-depth ACT. □

Element Distinctness

- Let W^- denote the set of negative instances of Element Distinctness:

$$W^- = \mathbb{R}^n \setminus W^+.$$

- How many connected components are there in W^-?

Application: Line Segment Intersection Detection

Theorem

Any ACT that solves the line segment intersection detection problem has height $\Omega(n \log n)$. Hence, the complexity of line segment intersection detection is $\Theta(n \log n)$ in the ACT model.

Proof.

Suppose T is an ACT that solves the line segment intersection detection problem. Let $(x_1, \ldots, x_n) \in \mathbb{R}^n$. We construct an ACT T' by plugging $(x_1, 0, x_1, 1, x_2, 0, x_2, 1, \ldots, x_n, 0, x_n, 1)$ to the input of T. That is, T' detects intersection between the segments $[(x_i, 0), (x_i, 1)]$. So T' solves the element distinctness problem. Thus $\mathrm{height}(T') = \Omega(n \log n)$. But by construction, $\mathrm{height}(T') = \mathrm{height}(T) + 4n$. Therefore $\mathrm{height}(T) = \Omega(n \log n)$. □

Concluding remarks

- We have seen a topological lower bound technique that shows that our line segment intersection detection algorithm from previous lecture is optimal.
- In next lecture, we will see more applications of this technique to computational geometry.