CS 516 -5
Computational Geometry & Graph Drawing
(Spring 2013)

Reading

MountNotes Chapters 7,8
Chapters 9,10, 6

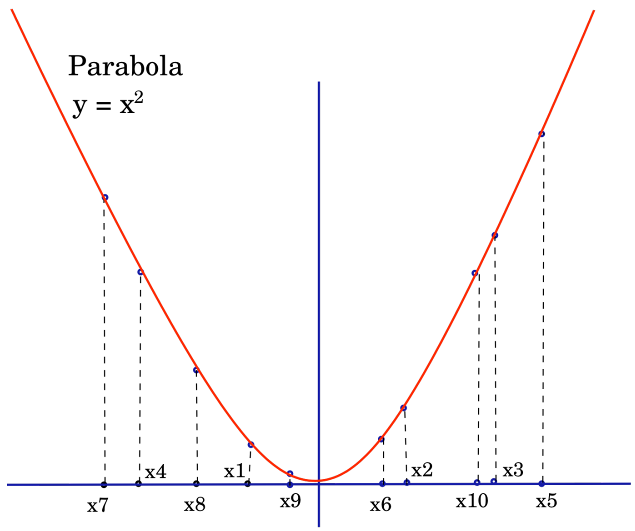
Last time…

- more convex hull algorithms (exploiting possible input-size output-size discrepancy)
  - More careful analysis of existing algorithms
    - E.g. unordered divide-and conquer
  - Try to discard non-extreme points quickly
    - Quickhull
  - "wrap" around the extreme points
    - Jarvis O(nh)

Last time (cont.)…

- more convex hull algorithms (exploiting possible input-size output-size discrepancy)
  - Marriage before conquest (K&S) *O(n lg h)*
    - Find bridge and filter *before* recursing
  - Chan's algorithm *O(n lg h)*
    - Clever combination of Jarvis and Graham
    - Successive "guesses" of size of *h*
- matching lower bounds
  - Decision tree framework

Lower Bounds



- **Reduce sorting to convex hull.**
- **List of numbers to sort $\{x_1, x_2, \ldots, x_n\}$.**
- **Create point $p_i = (x_i, x_i^2)$, for each i.**
- **Convex hull of $\{p_1, p_2, \ldots, p_n\}$ has points in sorted x-order. $\Rightarrow$ CH of n points must take $\Omega(n \log n)$ in worst-case time.**
- **More refined lower bound is $\Omega(n \log h)$. LB holds even for identifying the CH vertices.**

Lower bounds revisited

- recall limitations of the reduction from sorting argument
  - need a stronger model than pairwise comparisons

Lower bounds revisited

- recall limitations of the reduction from sorting argument
    - need a stronger model than pairwise comparisons
        - algebraic decision trees

Lower bounds revisited

- recall limitations of the reduction from sorting argument
  - need a stronger model than pairwise comparisons
    - algebraic decision trees
  - applies to strong version of CH problem (requires ordered output)
  - does not explain output-size sensitivity (dependence on h)

Lower bounds revisited

- recall limitations of the reduction from sorting argument
  - need a stronger model than pairwise comparisons
    - algebraic decision trees
  - applies to strong version of CH problem (requires ordered output)
  - does not explain output-size sensitivity (dependence on h)
    - formulate decision problems as point-classification problems

Lower bounds using fixed-order algebraic decision trees

Lower bounds using fixed-order algebraic decision trees

- Milnor's theorem
  - bounds the number of connected components of a region defined by common intersection of degree d surfaces
- Ben-Or's theorem
  - applies this to algebraic decision trees

Lower bounds using fixed-order algebraic decision trees

- Applications
  - element distinctness
  - multiset size verification
  - convex hull size verification

Loose ends from earlier discussions

- half-space intersection problem
  - how do we find a point in the common intersection (if it exists) in general?

Loose ends from earlier discussions

- half-space intersection problem
  - how do we find a point in the common intersection (if it exists) in general?
    - LP feasibility

Loose ends from earlier discussions

- half-space intersection problem
  - how do we find a point in the common intersection (if it exists) in general?
    - LP feasibility
- the marriage-before-conquest convex hull algorithm
  - need to find an (upper) bridge between opposite partitions. How do we do this efficiently?

Loose ends from earlier discussions

- half-space intersection problem
  - how do we find a point in the common intersection (if it exists) in general?
    - LP feasibility
- the marriage-before-conquest convex hull algorithm
  - need to find an (upper) bridge between opposite partitions. How do we do this efficiently?
    - a 2 variable (2-d) linear programming problem

Low-dimensional linear programming

- [2-d] a deterministic linear time algorithm
  - view as bridge-finding; candidate elimination
  - general LP formulation (Megiddo)
  - linear-time algorithms in higher dimensions

Low-dimensional linear programming (cont.)

- an incremental approach
  - in 1-d
  - in 2-d
    - (worst-case) analysis of deterministic implementation
    - (expected-case) analysis of randomized implementation

Low-dimensional linear programming (cont.)

- extensions to higher dimensions
  - Meggido's approach
  - randomized incremental approach

Applications

- 1-center problem…
  - “pinned” subproblems
    - uniqueness of solutions
    - reductions
- other LP-type problems

3-d convex hulls

- divide and conquer
- a “kinetic” approach