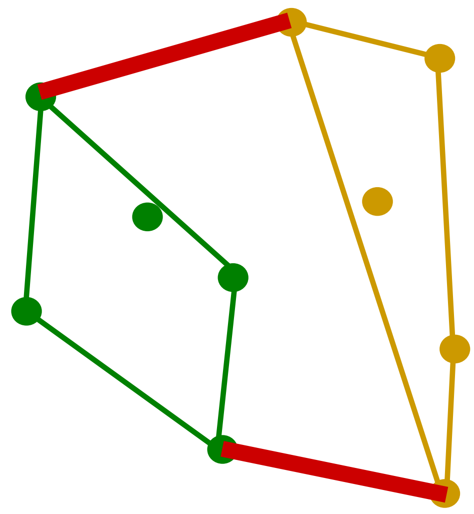
CS 6463: AT Computational Geometry
Fall 2010



Convex Hulls II

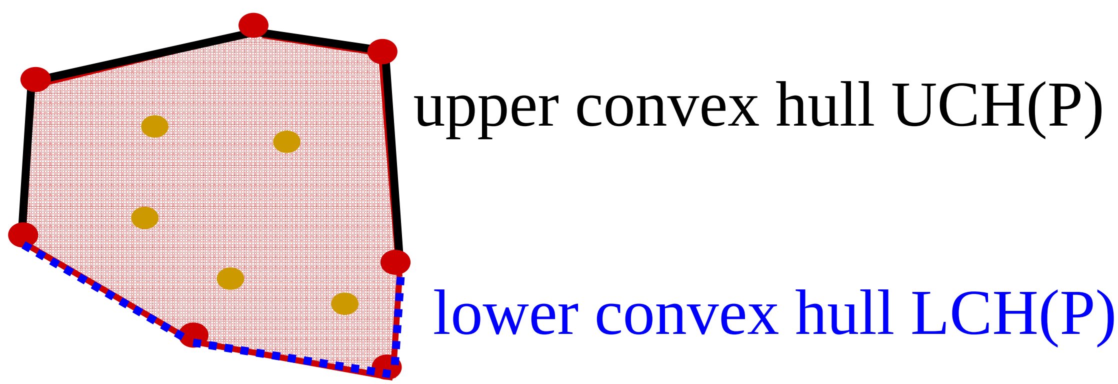
Carola Wenk

Graham's Scan

Incremental algorithm

- Compute solution by incrementally adding points
- Add points in which order?
  - Sorted by x-coordinate
  - But convex hulls are cyclically ordered

  → Break convex hull in **upper** and **lower** part



Graham's LCH

O(n log n)

O(n)

Algorithm Grahams_LCH(P):
// Incrementally compute the lower convex hull of P
Input: Point set $P \subseteq \mathbf{R}^2$
Output: A list L of vertices describing LCH(P) in counter-clockwise order

```
Sort P in increasing order by x-coordinate → P = {p_1,…,p_n}
L = {p_2,p_1}
for i=3 to n
    while |L|>=2 and orientation(L.second(), L.first(), p_i,) <= 0 // no left turn
        delete first element from L
    Append p_i to the front of L
```

- Each element is appended only once, and hence only deleted at most once $\Rightarrow$ the for-loop takes O(n) time
- O(n log n) time total

Lower Bound

- Comparison-based sorting of n elements takes $\Omega(n \log n)$ time.
- How can we use this lower bound to show a lower bound for the computation of the convex hull of n points in $\mathbf{R}^2$?
- Devise a sorting algorithm which uses the convex hull and otherwise only linear-time operations
  - $\Rightarrow$ Since this is a comparison-based sorting algorithm, the lower bound $\Omega(n \log n)$ applies
  - $\Rightarrow$ Since all other operations need linear time, the convex hull algorithm has to take $\Omega(n \log n)$ time

CH_Sort

Algorithm CH_Sort(S):

/* Sorts a set of numbers using a convex hull algorithm.
Converts numbers to points, runs CH, converts back to sorted sequence. */

Input: Set of numbers $S \subseteq \mathbf{R}$

Output: A list L of of numbers in S sorted in increasing order

P=∅

for each $s \in S$ insert (s,s^2) into P

L' = CH(P) // compute convex hull

Find point $p' \in P$ with minimum x-coordinate

for each $p=(p_x,p_y) \in L'$, starting with p', add p_x into L

return L