# COMP5045 Computational Geometry

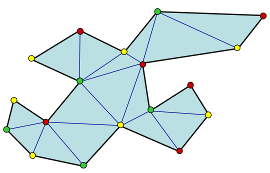


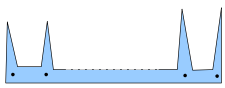




## Last week

- Every simple polygon with n vertices can be decomposed into n-2 triangles.
- Every triangulated simple polygon can be 3-colourable.
- Every simple polygon can be "guarded" by n/3 guards, and n/3 guards is sometimes necessary.
- To find a guard set our algorithm requires a triangulation.
  Last week: $O(n^2)$ time



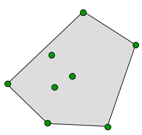






## Convex hull

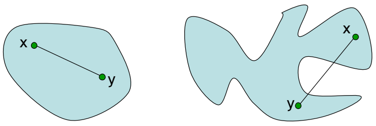
# Convex hulls and the sweep line technique



The imagination driving Australia's ICT future.

## Definition

A subset S of the plane is convex if for every pair of points x,y in S the straight line segment xy is completely contained in S.

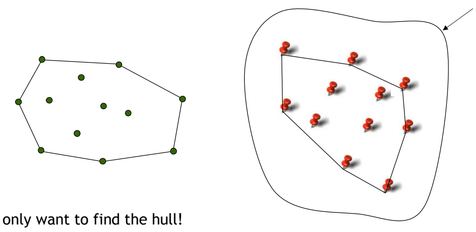





The imagination driving Australia's ICT future.

## Definition

The convex hull of a point set S is the smallest convex set containing S.




We only want to find the hull!



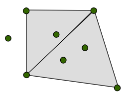
The imagination driving Australia's ICT future.

## CH algorithm - 1

Definition:
The CH of a set S of points in d dimensions is the union of all convex combinations of (d+1) points of S.

d=2: Convex combination of 3 points ⇒ a triangle!

Definition implies an algorithm:
A point that does not lie within any triangle of S is a CH vertex.



The imagination driving Australia's ICT future.

## CH algorithm - 1

Algorithm CH1(S)

```
1. for every possible triple of points x,y,z in S do
2.     for every point p in S do
3.        if p lies within the triangle (x,y,z) then
4.              discard p from S
```

Time complexity?

Step 1 is performed $O(n^3)$ times
Step 2 is performed n times/iteration
Step 3 and step 4 cost O(1) /iteration

Total time: $O(n^4)$

7

The imagination driving Australia's ICT future.

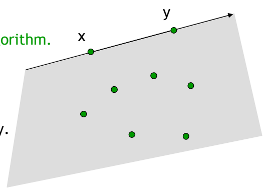
## CH algorithm - 2

Definition:
The CH of S is the intersection of all halfspaces that contain S.

This definition implies a second algorithm.

Consider an edge xy of CH(S).

All points of S must lie to the right of the directed line through x and y.



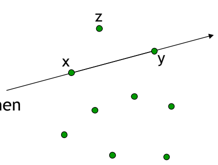

8

The imagination driving Australia's ICT future.

## CH algorithm - 2

Algorithm CH2(S)

```
1. for every ordered pair x,y in S do
2.     valid ← true
3.     for every point z in S-{x,y} do
4.           if z lies to the left of xy then
5.              valid ← false
6.     if valid then
7.        add xy to CH
8. Sort the edges in CH
```




Time complexity?

Steps 1-2, 6-7 : $O(n^2)$ times
Steps 3-5 : (n-2) times/iteration
Step 8 : O(n log n)

Total time: $O(n^3)$

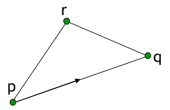
9

The imagination driving Australia's ICT future.

## Check left turn a primitive?

How can we check if a point r lies to the left of a line pq?

$\Rightarrow$ Triangle $\Delta(p,q,r)$ is oriented counter-clockwise.




$p=(p_x,p_y)$, $q=(q_x,q_y)$ and $r=(r_x,r_y)$

$$CCW(p,q,r) = \begin{vmatrix} p_x & q_x & r_x \\ p_y & q_y & r_y \\ 1 & 1 & 1 \end{vmatrix}$$

$$= (q_x - p_x)(r_y - p_y) - (r_x - p_x)(q_y - p_y)$$ [ 2 multiplications, 5 subtractions ]

$\Delta(p,q,r)$ is oriented counter-clockwise iff $CCW(p,q,r) > 0$.

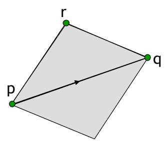
10

The imagination driving Australia's ICT future.

## Check left turn a primitive?

What is CCW(p,q,r)?

$|CCW(p,q,r)| =$




If pqr is a left turn then $CCW(p,q,r) > 0$

If pqr is a right turn then $CCW(p,q,r) < 0$

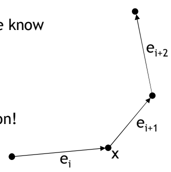
What about rounding errors?

11

The imagination driving Australia's ICT future.

## CH algorithm 3

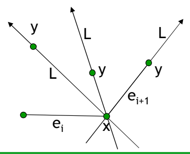
Can we compute the CH faster? Is there anything we know about the CH that we haven't used?




The edges in the CH are linked into a convex polygon!

If we found an edge on the CH with endpoint at x then the next edge must start at x.

Idea:
Draw a line L through x and a point y. Are there any points to the right of L?
If not (x,y) is an edge of CH.




Start point?

12

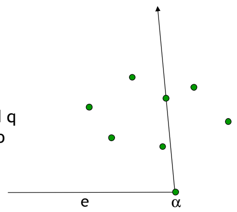
The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

## CH algorithm 3

Algorithm CH3(S)

1. find lowest point p in S
2. α ← p
3. repeat
4. for every point q in S-{α} do
5. valid ← true
6. L ← directed line through α and q
7. for every point r in S-{α,q} do
8. if r to the right of L then
9. valid ← false
10. if valid then
11. add αq to CH
12. α ← q
13. until α == p




Time complexity: $O(n^3)$



The imagination driving Australia's ICT future.

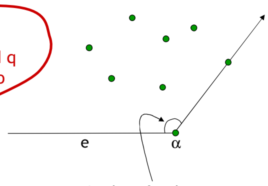
NATIONAL ICT AUSTRALIA

## CH algorithm 3

Algorithm CH3(S)

1. find lowest point p in S
2. α ← p
3. repeat
4. for every point q in S-{α} do
5. valid ← true
6. L ← directed line through α and q
7. for every point r in S-{α,q} do
8. if r to the right of L then
9. valid ← false
10. if valid then
11. add αq to CH
12. α ← q
13. until α == p

Can this be done faster?




Looking for the point with largest angle!



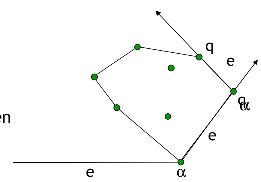
The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

## CH algorithm 4

Algorithm CH4(S)

1. find lowest point p in S
2. α ← p
3. e ← ( $(-\infty, p_y)$ , p )
4. repeat
5. maxAngle ← 0
6. for every point q in S-{α} do
7. if ∠( e , (α,q) ) > maxAngle then
8. nextPoint ← q
9. maxAngle ← ∠( e , (α,q) )
10. e ← ( α , nextPoint)
11. α ← nextPoint
12. until α == p

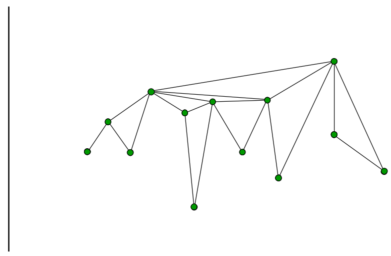



Time complexity: $O(n^2)$

What if the number of points on the CH is small?

## CH algorithm 5 - Incremental

Idea: Maintain hull while adding the points one by one, from left to right
⇔ sweep a line over the points from left to right





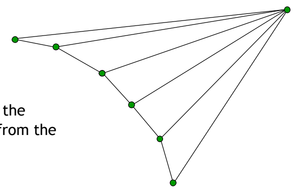


## CH algorithm 5 - Incremental

Running time?

O(n) per insertion ⇒ $O(n^2)$ in total

Can it be that bad?



A point is only added to the hull once and removed from the hull at most once!





## CH algorithm 5 - Incremental

Algorithm CH5(S)

1. sort the points in S from left to right $\langle p_1, p_2, \dots, p_n \rangle$
2. $L_{upper} \leftarrow \langle p_1, p_2 \rangle$
3. for $i \leftarrow 3$ to n do
4. append $p_i$ to $L_{upper}$
5. while $|L_{upper}| > 2$ and the last three points (a,b,c) turn left do
6. Delete b from $L_{upper}$
7. $L_{lower} \leftarrow \langle p_1, p_2 \rangle$

…

13. $L \leftarrow$ join($L_{upper}$, $L_{lower}$)
14. return L

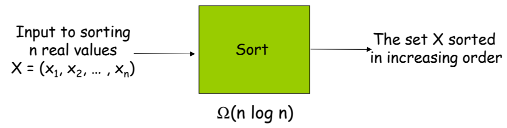
Time complexity: O(n log n)

## Lower bound

Can we do better than O(n log n)?

Prove a lower bound! Use a reduction from Sorting.

Sorting = Ω(n log n) in the algebraic decision tree model







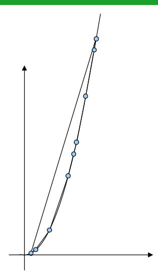

## Lower bound

For each value $x_i$ in X construct a point
$p_i = (x_i, x_i^2)$

P = ($p_1$, $p_2$, … , $p_n$)

Compute CH of P

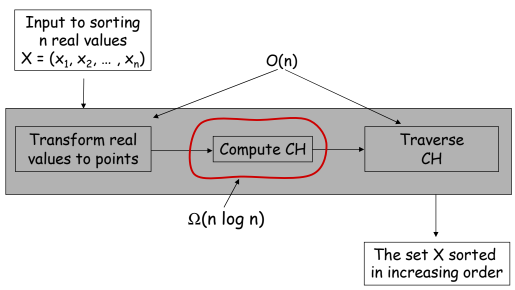
Find the leftmost point p in the CH.
Traverse the CH counter-clockwise from p and output the vertices in the order they are encountered. -> Points in sorted order!







## Lower bound



## What is known?

Preparata & Hong''77 O(n log n)

Kirkpatrick & Seidel'86 O(n log h)

Dynamic convex hull
Brodal & Jacob'02 O(log n) time/update

[This was an open problem since 1981.]

d dimensions
Chazelle'93 $\Theta(n^{\lfloor d/2 \rfloor})$ (complexity of CH)