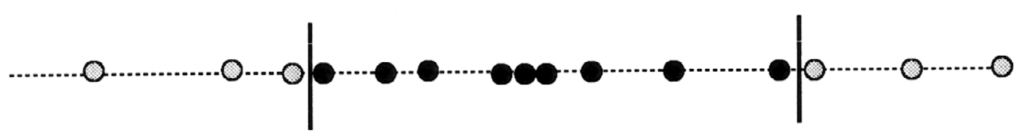
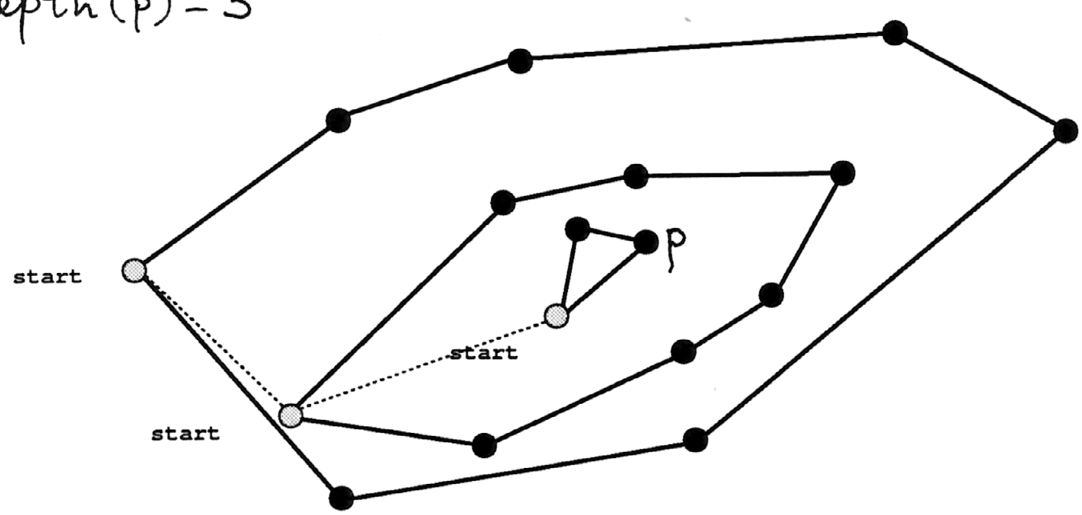
Robust Mean Estimation

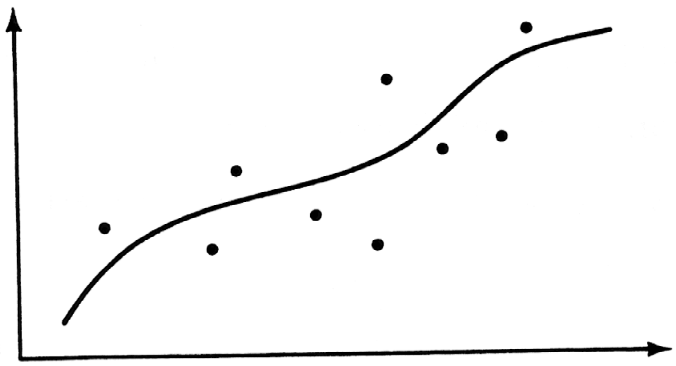
- $d = 1$: remove αn points from each side. Determine the mean
$$T = \frac{\sum_{i=\alpha n}^{(1-\alpha)n} x_i}{(1-2\alpha)n}$$
- $d = 2$: remove $2\alpha n$ points from outermost layers (one by one).
- Trivial algorithm $O(n^2 \log n)$
- Jarvis march $O(n^2)$
- Overmars and van Leeuven algorithm $O(n(\log n)^2)$
- Optimal algorithm exists $\Theta(n \log n)$



depth(p) = 3

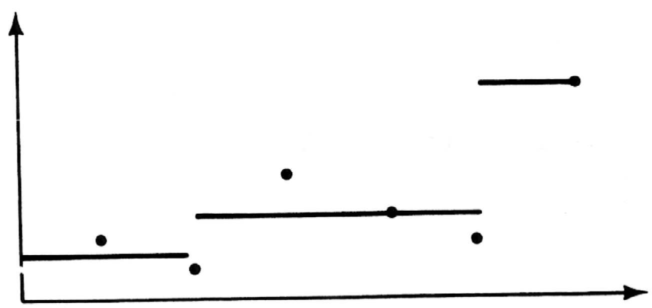


Isotonic regression



An isotone function (not best) approximating a finite point set.

$$\|f - f^*\| \to \min$$



A best isotone fit is a step function. The number of steps and the points at which they break must both be determined.

Isotonic Regression

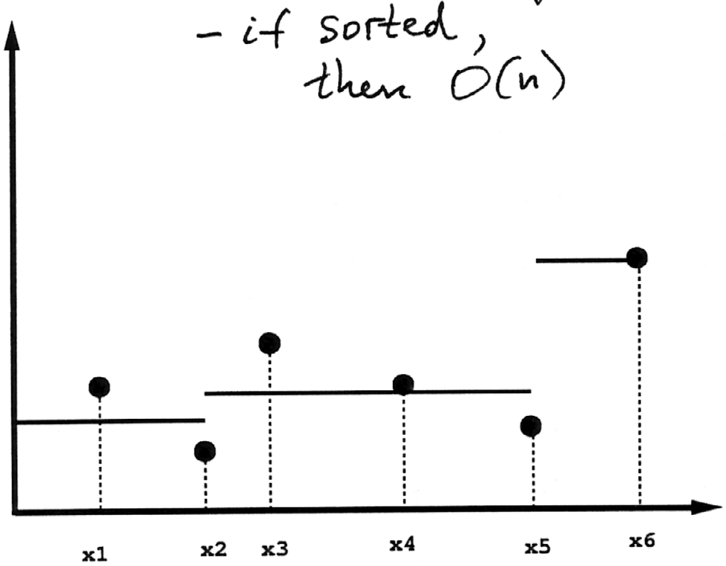
- **Given:** n points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ in the plane.
- **Find:** a non-decreasing step-function f minimizing

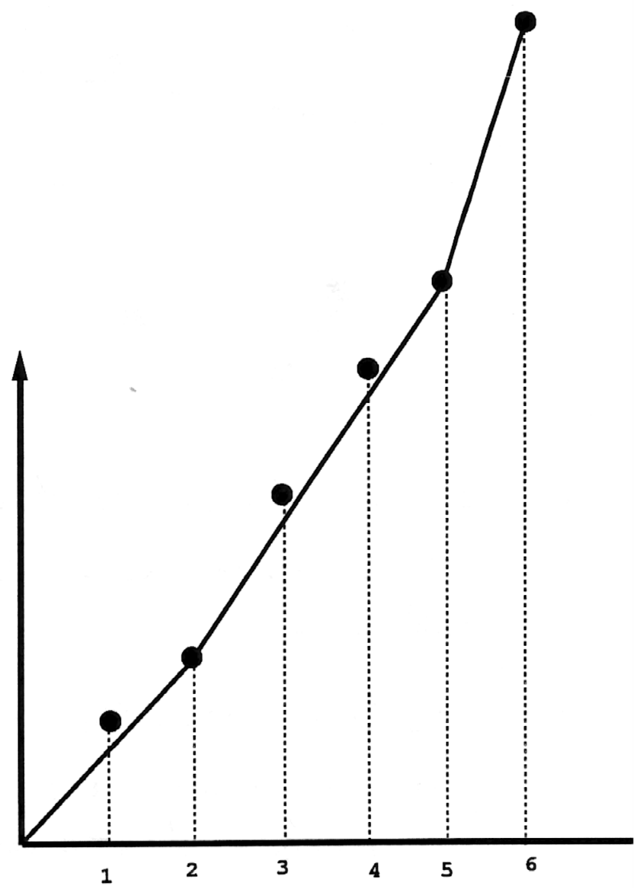
$$\sum_{i=1}^{n}(y_i - f(x_i))^2$$

- Jumps occur at $x_1, x_2, ..., x_n$ only.
- Assume consecutive jumps at x_i and x_k, $i < k$. Then $f(x_p) = (\sum_{l=i}^{k-1} y_l)/(k-i)$.
- The slope of the tangent to the convex hull under p_i is the value of $f(x_i)$.

$p_0 = (0,0)$,
$p_j = (j, \sum_{i=1}^{j} y_i)$

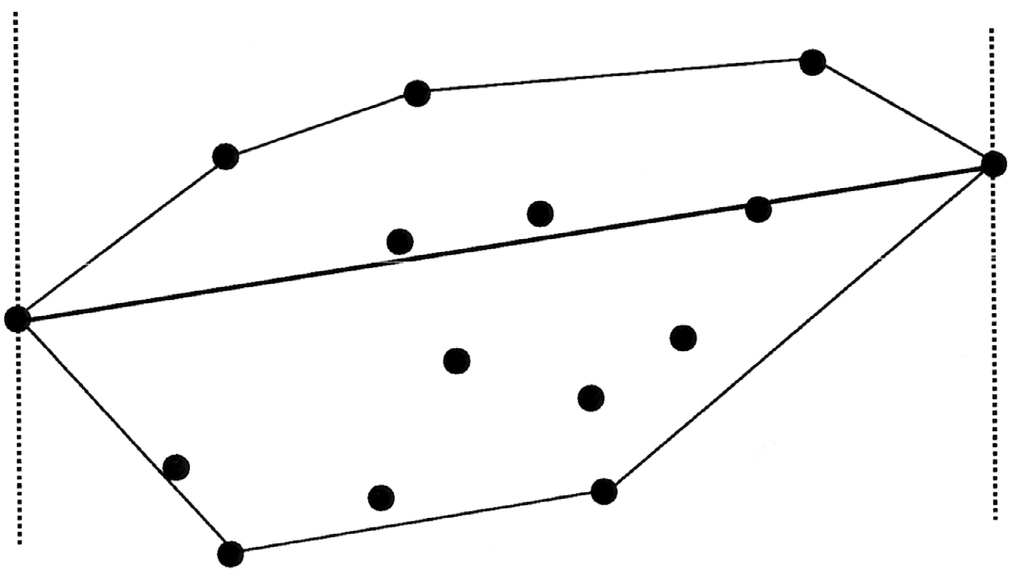
- $f(x)$ in $O(n \log n)$
  - if sorted, then $O(n)$



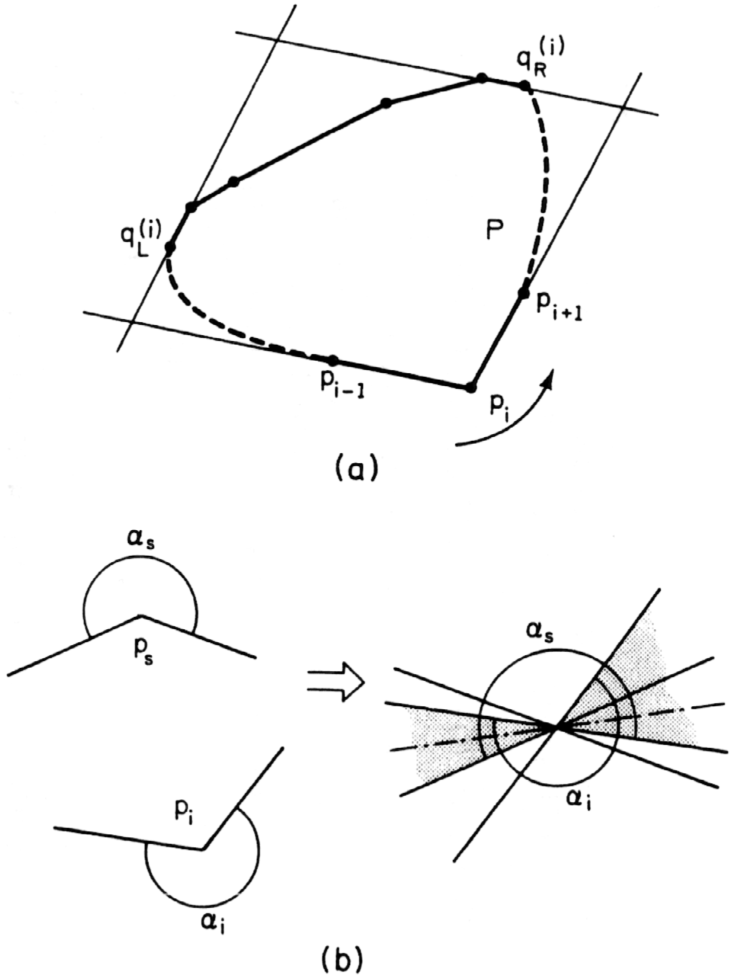


Diameter of a Point Set

- **Given:** n points in the plane.
- **Find:** Pair of points farthest apart.
- Diameter of a point sets is equal to the diameter of its convex hull.
- Distance between extreme points of a convex set is greatest between parallel supporting lines.



- We need to solve this problem many times in the minimum diameter k-clastering problem
- Yaglom-Boltyanskii (1961)
  - The diameter of a convex figure is the greatest distance between parallel lines of support.



Characterization of antipodal pairs. The intersection of α_i and α_s is a pair of planar wedges sharing their vertex.

- a pair of points that does admit parallel supporting lines is called <u>antipodal</u>.

Code and Example

```
        procedure ANTIPODAL PAIRS
1.  begin p := p_N;
2.      q := NEXT[p];
3.      while (Area(p, NEXT[p], NEXT[q]) > Area(p, NEXT[p], q)) do
            q := NEXT[q];
        (*march on P until you reach the first vertex farthest from
        pNEXT[p]*)
4.      q_0 := q;
5.      while (q ≠ p_0) do
6.        begin p := NEXT[p];
7.              print (p, q);
8.              while (Area(p, NEXT[p], NEXT[q]) > Area(p,
                    NEXT[p], q)) do
9.                begin q := NEXT[q];
10.                     if ((p, q) ≠ (q_0, p_0)) then print (p, q)
                  end;
11.             if (Area(p, NEXT[p], NEXT[q]) = Area(p, NEXT[p], q))
                  then
12.                 if ((p, q) ≠ (q_0, p_N)) then print (p, NEXT[q])
                    (*handling of parallel edges*)
        end
    end.
```

p	q	Print	Step
p_9			1
	p_0		2
	p_1		3
	p_2		3
	p_3		3
	p_4		3
	p_5		3
p_0			6
		$(p_0\ p_5)$	7
p_1			6
		$(p_1\ p_5)$	7
	p_6		9
		$(p_1\ p_6)$	10
		$(p_1\ p_7)$	12
p_2			6
		$(p_2\ p_6)$	7
	p_7		9
		$(p_2\ p_7)$	10
	p_8		9
		$(p_2\ p_8)$	10
	p_9		9
		$(p_2\ p_9)$	10
p_3			6
		$(p_3\ p_9)$	7
p_4			6
		$(p_4\ p_9)$	7
p_5			6
		$(p_5\ p_9)$	7
	p_0		9

While Loop

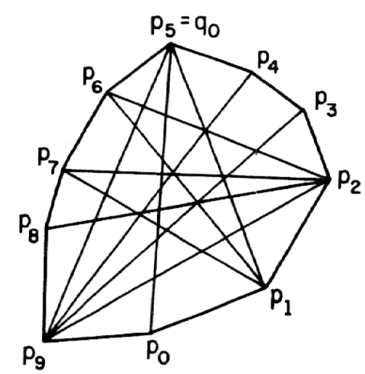


Illustration of ANTIPODAL PAIRS. Note that $\overline{p_1 p_2}$ and $\overline{p_6 p_7}$ are parallel edges.

Diameter of a Point Set - Lower Bound

- **Given:** Two sets $A = \{a_1, ..., a_n\}$ and $B = \{b_1, ..., b_n\}$ of n real numbers.
- Deciding whether $A \cap B = \emptyset$ requires $\Theta(n \log n)$ time.
- Consider a circle C with center in the origo. Map each number a_i to the intersection point of the line $y = a_i x$ with C (in the first quadrant). Map each number b_i to the intersection point of the line $y = b_i x$ with C (in the third quadrant).
- Let S denote the set of $2n$ points.
- $A \cap B = \emptyset$ iff $diam(S) = diam(C)$.