# $k$-smallest Element

## by Randomization

C. A. R. Hoare; *Algorithm 65: find*, CACM 4:321–322, 1961.

RandomizedSelect$(L, k)$

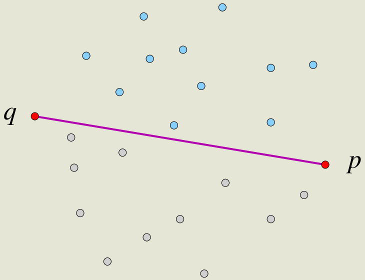
pick an element $r$ from $L$ at random<br>
$L_< \leftarrow \{x \in L \mid x < r\}$; $L_> \leftarrow \{x \in L \mid x > r\}$<br>
**if** $(k \leq |L_<|)$<br>
    **then return** RandomizedSelect$(L_<, k)$<br>
    **else if** $(k > n - |L_>|)$<br>
            **then return** RandomizedSelect$(L_>, k - n + |L_>|)$<br>
            **else return** $r$

# 2D Convex Hull

## by Divide, Prune & Conquer

## QuickHull

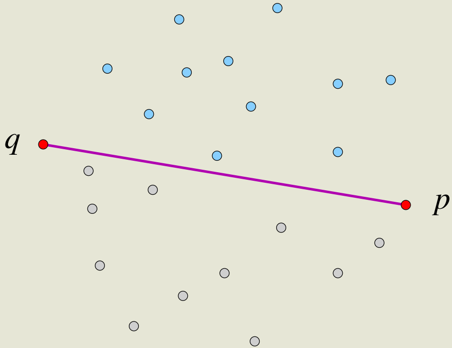
$$S = \{p_1, p_2, \ldots, p_n\} \subset \mathbb{R}^2$$

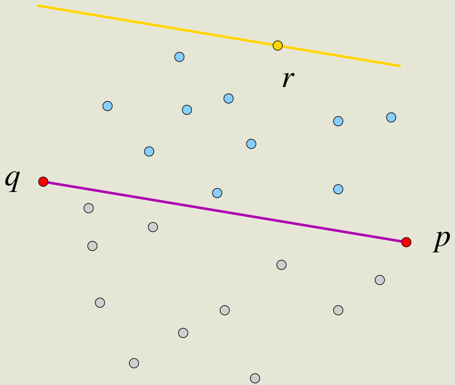


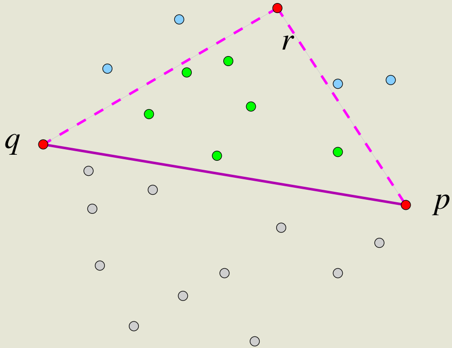
Points $p$ and $q$ are extreme points of $S$. The non-empty set $S_{pq} \subset S$ contains all extreme points that are counterclockwise between $p$ and $q$.

$\textsc{QuickSubhull}(p, q, S_{pq})$

**if** $S_{pq} = \emptyset$<br>
Then **return** $\overline{pq}$<br>
**if** $S_{pq} = \{r\}$<br>
Then **return** $\overline{pr}, \overline{rq}$<br>
$r \leftarrow$point furthest to the right of $\ell(p, q)$<br>
$S_{pr} \leftarrow \{s \in S_{pq} \mid s$ lies right of $\ell(p, r)$<br>
$S_{rq} \leftarrow \{s \in S_{pq} \mid s$ lies right of $\ell(r, q)$<br>
**return** $\textsc{QuickSubhull}(p, r, S_{pr}), \textsc{QuickSubhull}(r, q, S_{rq})$


q
p


r
q
p


r
q
p

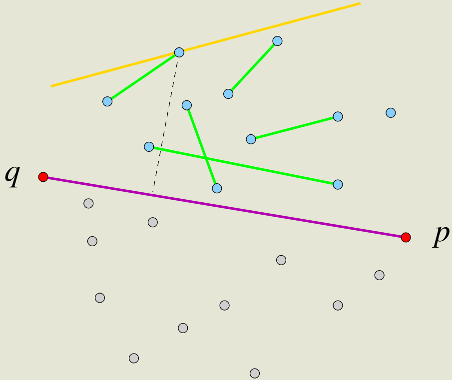
Observations:

- We find initial extreme points by computing the leftmost and rightmost points (lexicographically)
- In the worst-case, we cannot discard any points and get an totally unbalanced split: $|S_{pq}| - 1$ points are in one set while the other one is empty
- The resulting worst-case running time is $\Theta(n^2)$
- There is a recursive call only if we found a new extreme point

Theorem

*Let $S$ be a set of $n$ points in the plane with $h$ extreme points. Using* QuickSubhull *we can compute the convex hull of $S$ in time $O(nh)$.*

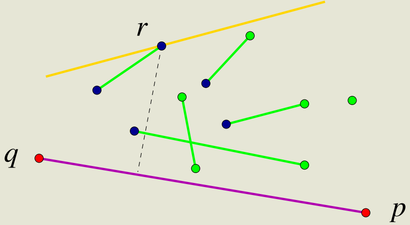
# FastHull



$\textsc{FastSubhull}(p, q, S_{pq})$

**if** $S_{pq} = \emptyset$<br>
Then **return** $\overline{pq}$<br>
**if** $S_{pq} = \{r\}$<br>
Then **return** $\overline{pr}, \overline{rq}$<br>
choose $\lfloor \frac{n}{2} \rfloor$ disjoint pairs of points in $S_{pq}$ and order pairs $(s,t)$<br>
such that $\textsc{proj}(t, \ell(p,q))$ is closer to $p$ than $\textsc{proj}(s, \ell(p,q))$<br>
$L \leftarrow$ set of lines formed by these pairs of points<br>
$\ell_{\text{med}} \leftarrow \textsc{MedianSlope}(L)$<br>
$r \leftarrow$ furthest point in $S_{pq}$ in direction orthogonal to $\ell_{\text{med}}$<br>
$r' \leftarrow \textsc{proj}(r, \ell(p,q))$<br>
$S_{pr} \leftarrow \{s \in S_{pq} \mid s$ lies right of $\ell(r', r)\}$<br>
$S_{rq} \leftarrow \{s \in S_{pq} \mid s$ lies left of $\ell(r', r)\}$

**for** all pairs $(s, t)$
    **do if** $\textsc{SmallerSlope}(\ell(s,t), \ell_{\text{med}})$
        **then** remove $t$ from $S_{rq}$
      **if** $\textsc{LargerSlope}(\ell(s,t), \ell_{\text{med}})$
        **then** remove $s$ from $S_{pr}$




**if** $(r = q)$
  **then return** $\textsc{FastSubhull}(p, r, S_{pr})$
**if** $(r = p)$
  **then return** $\textsc{FastSubhull}(r, q, S_{rq})$
**return** $\textsc{FastSubhull}(p, r, S_{pr}), \textsc{FastSubhull}(r, q, S_{rq})$

Analysis:

Let $n = |S_{pq}|$ and let $n_1 = |S_{pr}|$ and $n_2 = |S_{rq}|$.

$$T(n,h) = T(n_1,h_1) + T(n_2,h_2) + O(n)$$

$$n_1 + n_2 < n$$

$$h_1 + h_2 \le h$$

## Lemma

$$n_i \le \frac{3}{4}n$$

Claim: $$T(n,h) \leq c_0 n \log h$$

$$\begin{aligned} T(n,h) &\leq c_0 n_1 \log h_1 + c_0 n_2 \log h_2 + c_1 n \\ &\leq c_0 n_i \log h/2 + c_0 n_{3-i} \log h_{3-i} + c_1 n \\ &\leq c_0 (n_1 + n_2) \log h - c_0 n_i + c_1 n \\ &\leq c_0 n \log h \end{aligned}$$

for $c_0 \geq 8c_1$, because either $n_1 + n_2 \leq \frac{7}{8}n$ or $n_i \geq \frac{1}{8}n$.

## Theorem

*Let $S$ be a set of $n$ points in the plane and let $h$ be the number of extreme points in $S$. Using* FastSubhull *we can compute the convex hull of $S$ in worst-case time $O(n \log h)$.*

# Overview

- geometric structures
    - convex hull
    - triangulation
    - Voronoi diagrams
    - Delaunay diagrams
    - arrangements
- geometric algorithm design paradigms
    - incremental construction
    - divide & conquer
    - plane-sweep
    - geometric transformations
    - randomized incremental construction
    - prune & search