Convex Hull

Quickhull

01-XB

Junhui DENG

deng@tsinghua.edu.cn

Quicksort

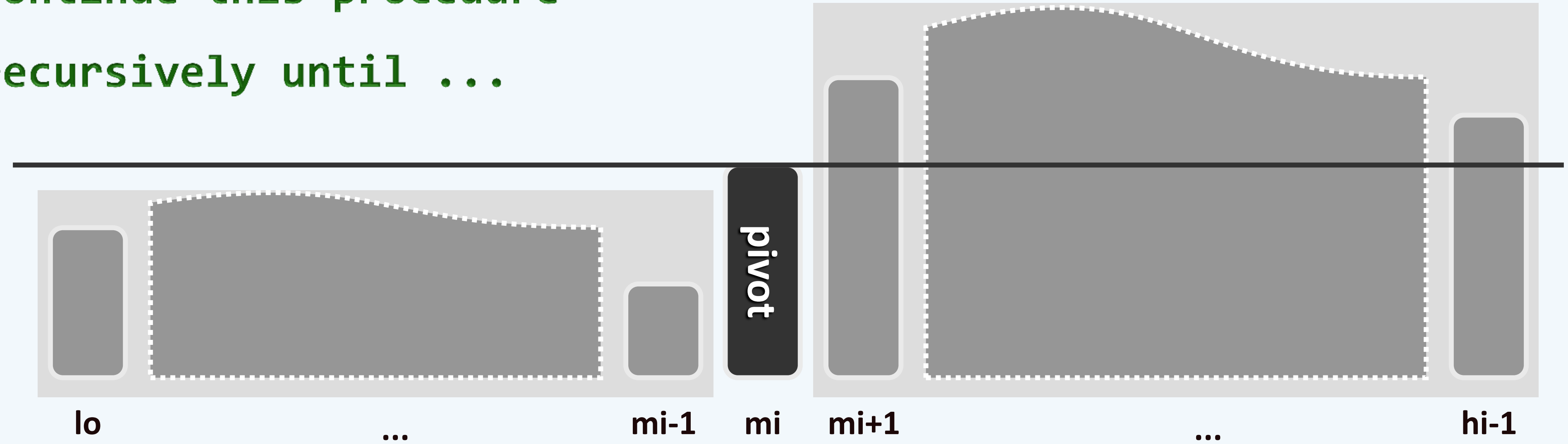
❖ C. A. R. Hoare, 1960

- $\boxed{\text{partition}}$ the input array A into 2 sub-arrays L and R s.t.

  $$\boxed{L \leq \text{pivot} \leq R}$$

- $\boxed{\text{partition}}$ L and R respectively in the same manner
- continue this procedure

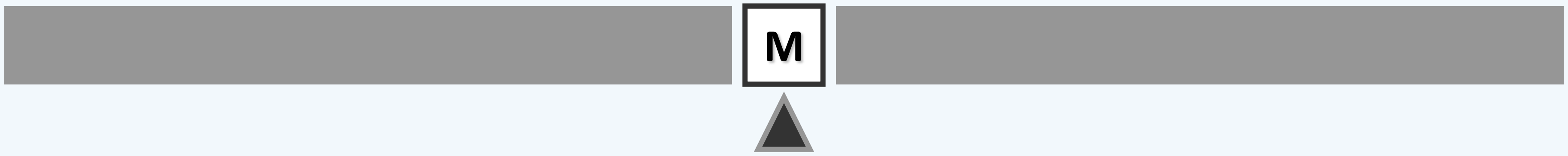
  recursively until ...



Pivot

- The most critical issue concerning efficiency of Quicksort is the pivot
- Which element is the best pivot? median!

  It partitions the array into 2 sub-arrays with almost equal sizes
- However, Quicksort needs $\mathcal{O}(n^2)$ time for worst inputs //examples please
- Fortunately, the average performance of Quicksort is good enough



Intuition

- In many CH applications,

  most of the input points lie interior to the hull

- Specifically, as we know,

  if the points are

  uniformly distributed in a square,

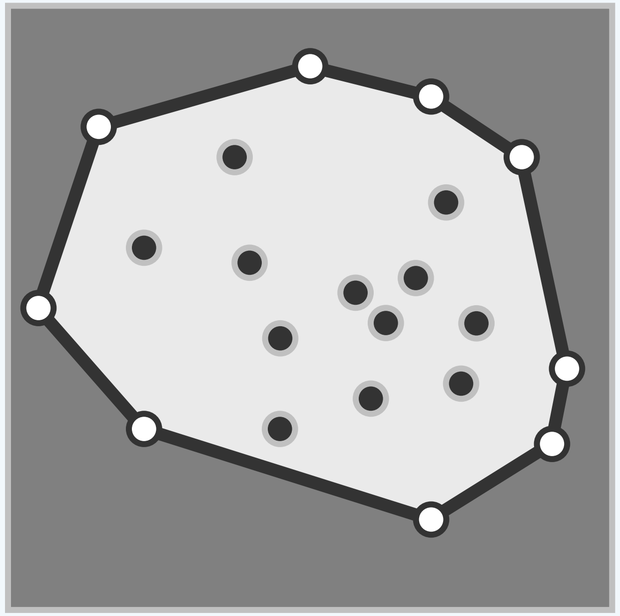
  the expected number of EP's is $\mathcal{O}(\log n)$

- In such kind of conditions,

  can we determine and exclude

  the non-extreme points

  even more quickly?



Upper & Lower Hulls

❖ Each convex hull has

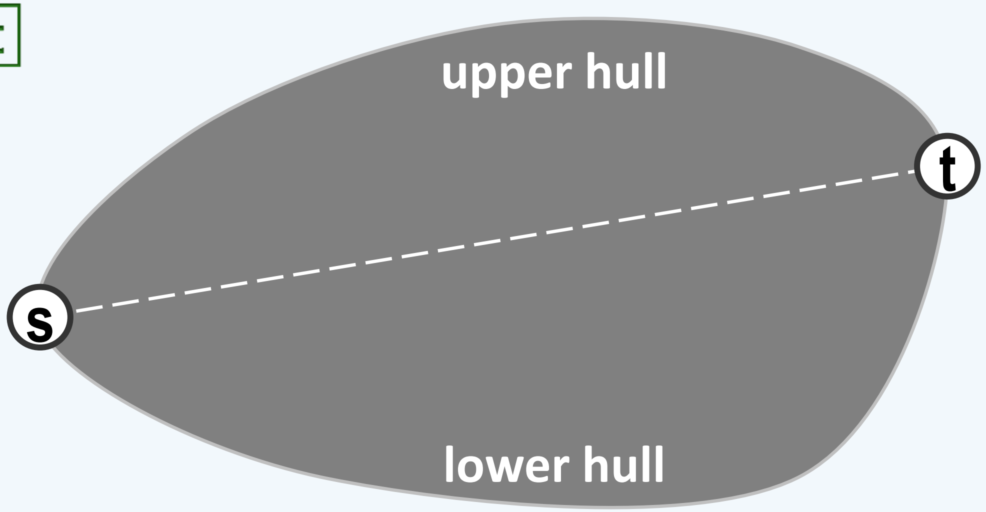
a unique $\boxed{\text{leftmost-then-lowest}}$/$\boxed{\text{rightmost-then-highest}}$ EP $\boxed{s}$/$\boxed{t}$

❖ Each convex hull is divided by $\boxed{s}$ and $\boxed{t}$

into 2 subhulls

the upper hull: U_CH(P)

the lower hull: L_CH(P)



❖ CH(P) can be computed

from U_CH(P) and L_CH(P) in $\boxed{\mathcal{O}(1)}$ time

❖ By symmetry

the construction of CH(P) can be $\boxed{\text{reduced}}$ to the construction of U_CH(P)

Computing U_CH(P)

- $\boxed{\text{Idea}}$

partition the input set P into 3 subsets $\boxed{P_0}$, $\boxed{P_1}$ and $\boxed{P_2}$ s.t.

1) P_0 doesn't intersect U_CH(P) and hence can be excluded

2) U_CH(P_1) and U_CH(P_2) will be computed recursively and their concatenation gives U_CH(P)

i.e., $\boxed{U\_CH(P) = U\_CH(P_1) \boxplus U\_CH(P_2)}$

- How would the partition be done?

Apex of the Roof

- It suffices to consider only those points lying $\boxed{\text{above}}$ the segment $\boxed{st}$

  Or, equivalently, those lying $\boxed{\text{left}}$ to the directed line $\boxed{st}$

- Consider the point, say $\boxed{r}$, with the $\boxed{\text{maximal}}$ distance to $\boxed{st}$

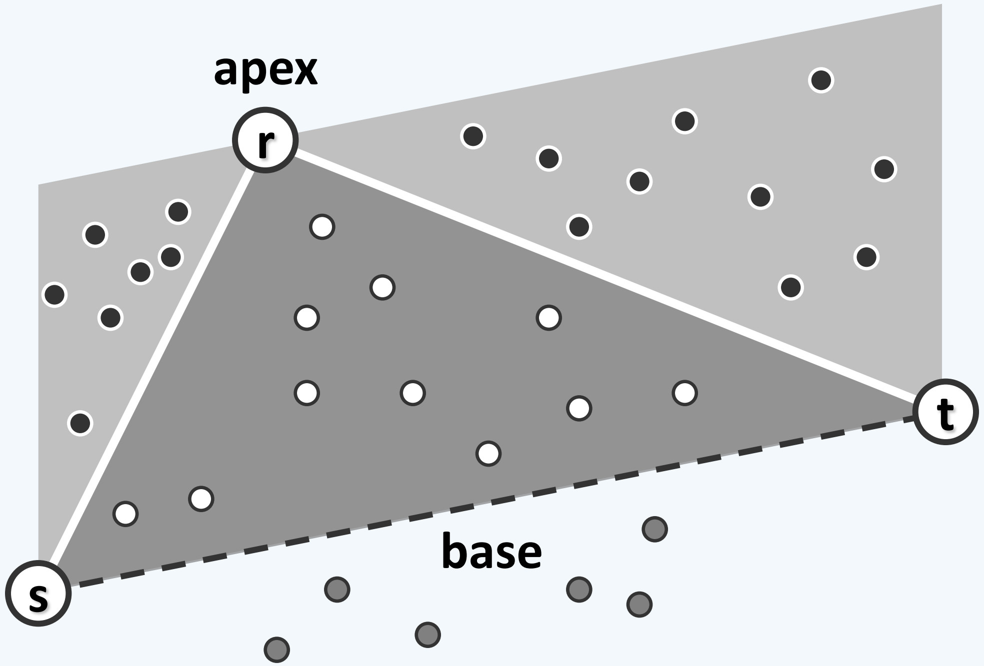
- Hence $\boxed{r}$ belongs to U_CH(P)

- The directed segments $\boxed{sr}$ and $\boxed{rt}$

  are called the $\boxed{\text{roof}}$ of P, with

  r the $\boxed{\text{apex}}$ and

  st the $\boxed{\text{base}}$



Divide-and-conquer

❖ Point set partition

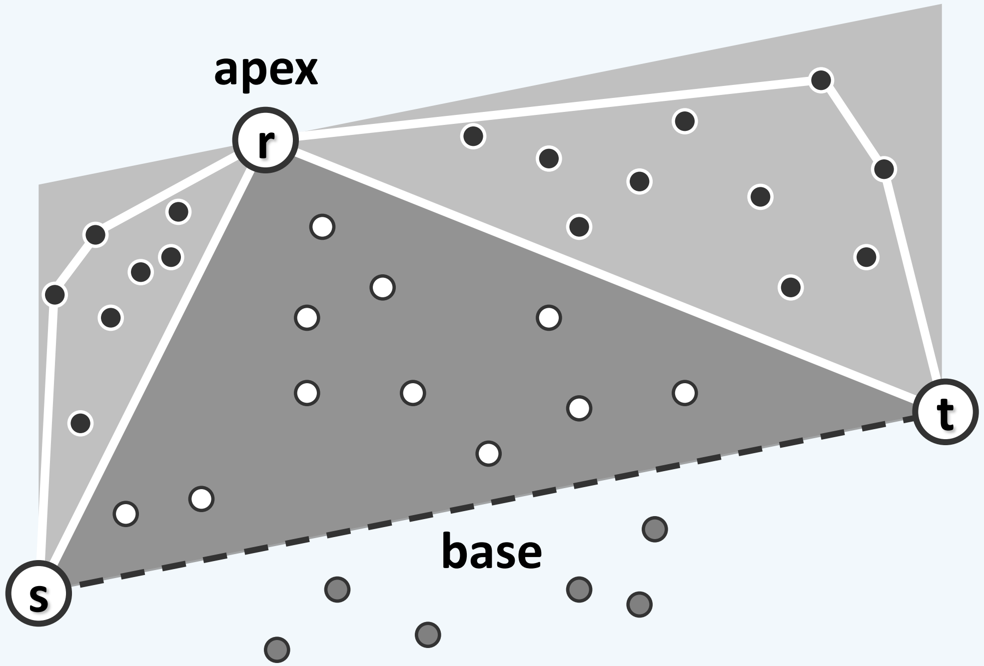
$\boxed{P_0}$: points $\boxed{\text{below}}$ the $\boxed{\text{roof}}$

$\boxed{P_1}$: points left to the directed $\boxed{sr}$

$\boxed{P_2}$: points left to the directed $\boxed{rt}$

❖ Could $\boxed{P_1}$ intersect with $\boxed{P_2}$?

Why?

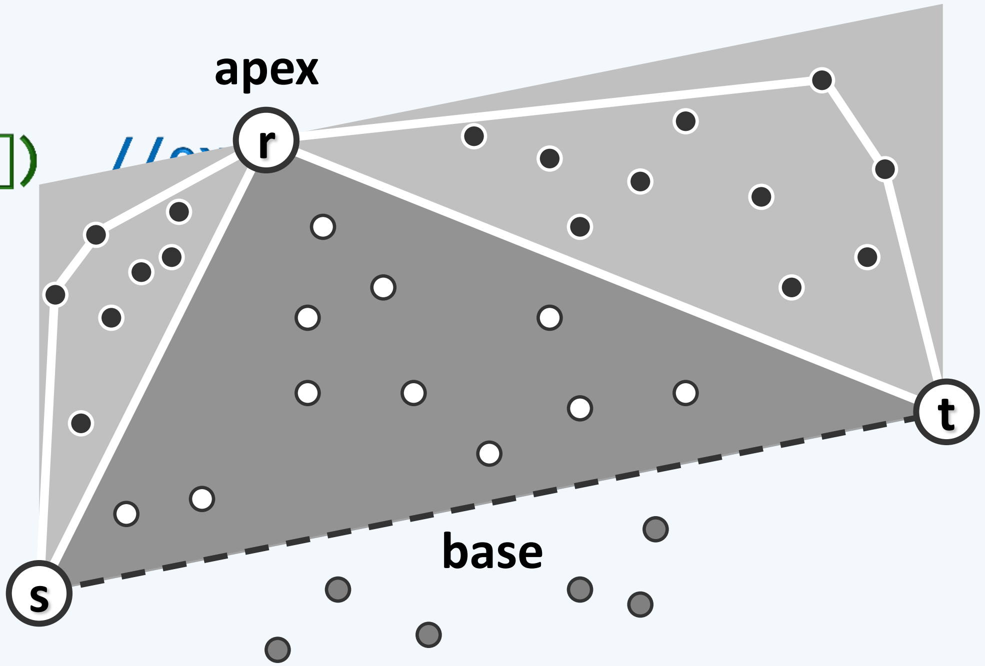


Time Cost

- Generally, subset $\boxed{P_0}$ has $\boxed{\text{much more}}$ points than the other two s.t.
  1) the number of points to be considered will decrease tremendously and
  2) a small depth of recursion will be achieved
- Best case: $\mathcal{O}(\boxed{n})$ //example ...
- Worst case: $\mathcal{O}(\boxed{n^2})$ //e
- Average case: depends on
  1) how $\boxed{\text{evenly}}$ the points

     are distributed and

  2) ...



Further readings

❖ Randomized Quickhull

1) R. Wenger

Randomized Quick Hull

Algorithmica, 17 (1997), pp. 322-329

2) C. B. Barber, D. P. Dobkin, H. Huhdanpaa

The Quickhull Algorithm for Convex Hulls

ACM Trans. Mathematical Software, 22 (1996), pp. 469-483