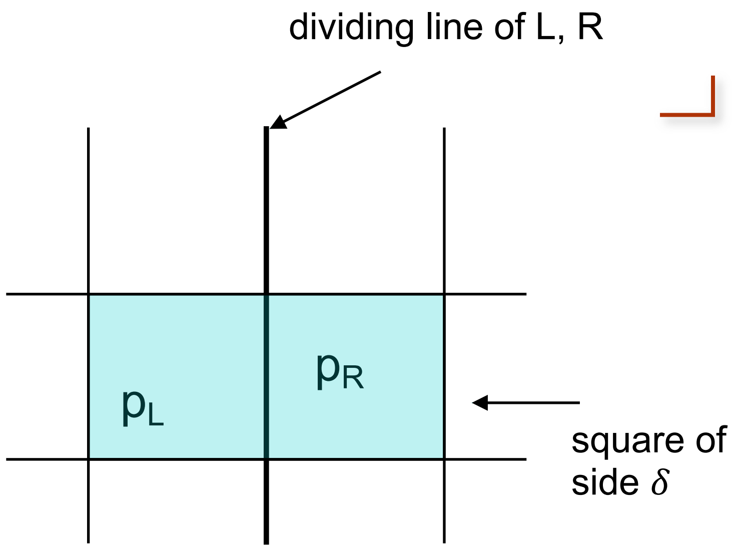
Closest pair of points in 2D

- Naïve solution takes $O(n^2)$.
- Apply D&C to reduce complexity to $O(n \log n)$.
- Recurrence: $T(n) = 2T(n/2) + O(n)$
- Initialization step:
  - Sort points by X
  - Sort points by Y
  - Partition points into L and R by X
  - Answer = min(closest within L, closest within R, closest across L and R)
- All the innovation goes into finding the closest pair across L and R.

Searching across L, R



- Let $\delta = \min(\delta_L, \delta_R)$
- Any closer pair of points must lie within a rectangle as shown above
- At most 4 points from L can lie within a δ square.
  - Same for R
- A total of at most 8 points can lie within a rectangle of Y-dim δ and X-dim 2δ.
- Sort all the points lying within δ of the dividing line by Y-value and examine in sorted order.
- How many neighboring points to examine for every point?
  - Answer is 7

Notes

Selecting item of rank i

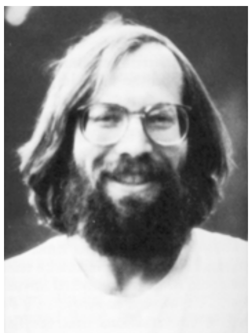
- Median of n items is the item with rank $\frac{n}{2}$.
- Rank of an item is its position in the list if the items were sorted in ascending order.
- Rank i item also called ith statistic.
- Popular statistics are quantiles: items of rank $\frac{n}{4}, \frac{n}{2}, \frac{3n}{4}$.
- After spending O(n log n) time on sorting, any rank can be found in O(n) time.
- Can we find a rank without sorting?

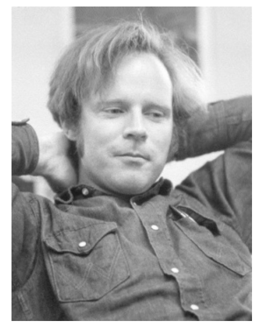
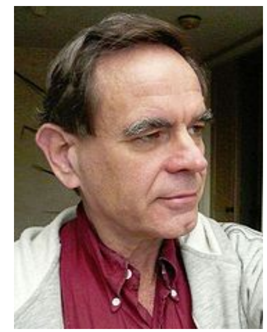
What is the time to find items of rank 1 or n?

How to apply Divide & Conquer?

BFPRT Algorithm

- A very clever choose algorithm . . .




Blum, M.; Floyd, R. W.; Pratt, V. R.; Rivest, R. L.; Tarjan, R. E. (August 1973). "Time bounds for selection" (PDF). *Journal of Computer and System Sciences*. **7** (4): 448–461.

Linear Time Selection

<u>$SELECT\ (i)$:</u>

1. Divide items into $\left\lceil\frac{n}{5}\right\rceil$ groups of 5 each.
2. Find the median of each group (using sorting).
3. Recursively find median x of $\left\lceil\frac{n}{5}\right\rceil$ group medians.
4. Partition using x as pivot into low and high. Let low side have k items and high side have $n - k - 1$ items.
5. Compare i, k
   - If $i = k$ then return x
   - If $i < k$ then call $SELECT(i)$ on low side
   - otherwise, call $SELECT(i - k)$ on high side

Assume distinct values

Analysis

- Half of $\left\lceil \frac{n}{5} \right\rceil$ groups contribute 3 elements smaller than x
  - Not counting the group with x and another possibly incomplete group, the number of elements adds up to $\frac{3n}{10} - 6$.
- Similarly, there are at least $\frac{3n}{10} - 6$ elements larger than x.
- SELECT is called recursively in Step 5 on at most $\frac{7n}{10} +$ 6 elements
- $T(n) \leq T\left(\left\lceil \frac{n}{5} \right\rceil\right) + \mathrm{T}\left(\frac{7n}{10} + 6\right) + O(n)$ if n $\geq$ 140

  $= \Theta(1)$, otherwise

A constant deciding when recursion bottoms out.

Analysis contd.

- Prove by induction that $T(n) \leq cn, \forall n$

The value of c will follow from the proof

  - constant c is chosen large enough so that $T(n) \leq cn$ for the base case
- For the O(n) term in the recurrence, choose bn
  - Choose b high enough to account for bookkeeping cost in the recursive step
- Applying the induction hypothesis, /* inductive step */

$$T(n) \leq c\left\lceil\frac{n}{5}\right\rceil + \mathrm{c}\left(\frac{7n}{10} + 6\right) + bn$$

$$\leq \frac{cn}{5} + c + \frac{7cn}{10} + 6c + bn$$

$$= \frac{9cn}{10} + 7c + bn$$

$$= cn + (-\frac{cn}{10} + 7c + bn)$$

$$\leq cn + (-\frac{cn}{10} + 7c + bn)$$

$$\leq cn \text{ provided } -\frac{cn}{10} + 7c + bn \leq 0, \text{ or } c \geq \frac{10bn}{n-70} = 10\mathrm{bn}\left(\frac{n}{n-70}\right)$$

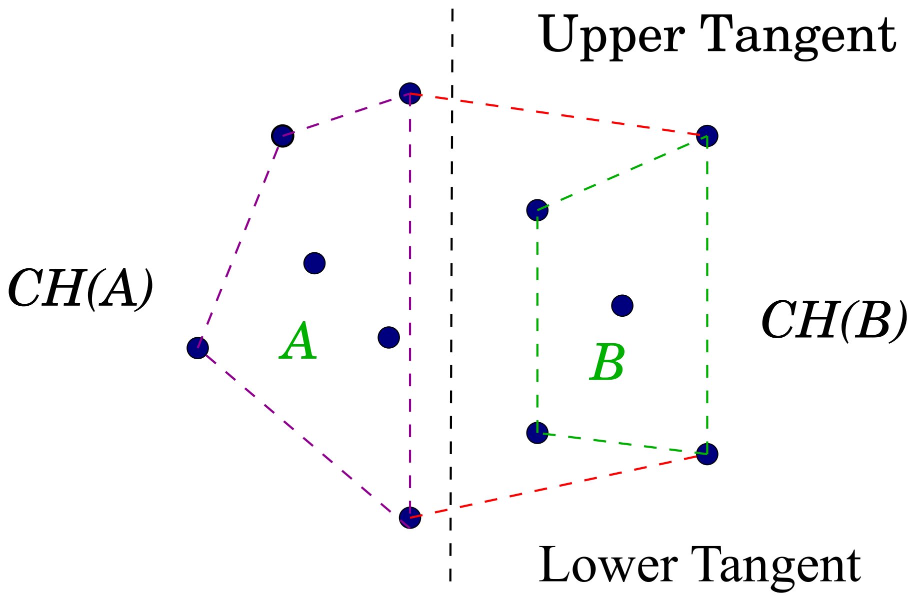
- Choose $c \geq 20b$ /*note that the ratio is at its highest value of 2 when n = 140 */
- Thus, $T(n) \leq cn$

Notes

Convex Hull

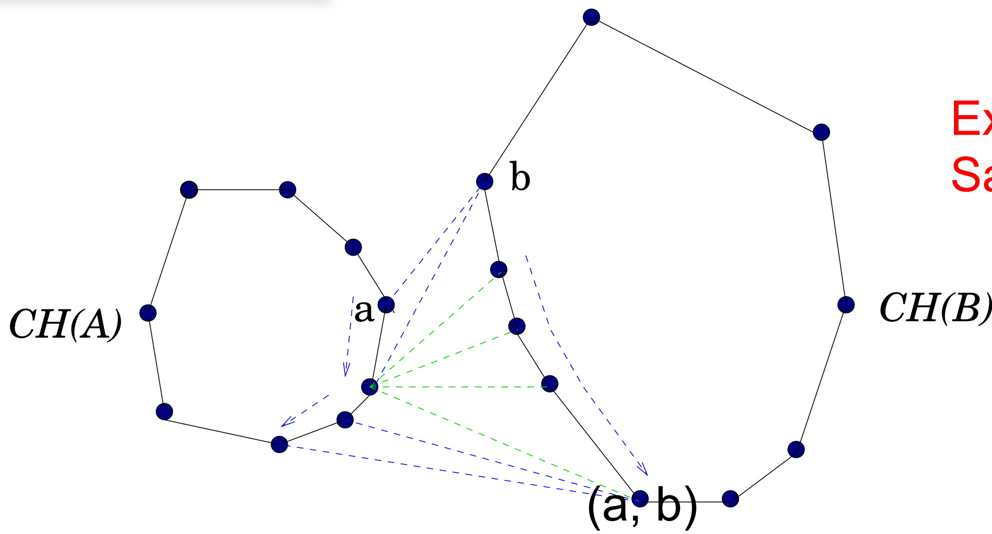
- Many applications in robotics, shape analysis, line fitting etc.
- Example: if the convex hulls of two objects are disjoint then the objects do not intersect.
- Given a set of points S, compute the convex hull CH**(**S**),** ordered set of vertices.

Divide and Conquer



- Sort points in S by x-coordinates.
- Divide points into equal halves A and B.
- Recursively compute CH(A) and CH(B).
- Merge CH(A) and CH(B) to obtain CH(S).

Tangent Finding for Merging



Example for lower tangent
Same idea for upper tangent

- a = rightmost point of CH(A), b = leftmost point of CH(B).
- while (a, b) not lower tangent of CH(A) and CH(B) do
- while (a, b) not lower tangent to CH(A)
  move a to next point on CH(A) clockwise
- while (a, b) not lower tangent to CH(B)
- move b to next point on CH(B) counterclockwise
- Return (a, b)

O(N) time

Analysis

- Initial sorting takes O(N log N) time.
- Recurrence: $T(N) = 2T(\frac{N}{2}) + O(N)$
- Solution: $T(N) = O(N \log N)$