Algorithm Design and Analysis

CSE 565

LECTURE 13

Divide and Conquer

- Closest Pair of Points

Sofya Raskhodnikova

9/26/2007

S. Raskhodnikova; based on slides by E. Demaine, C. Leiserson, A. Smith, K. Wayne

Review questions

- Guess the solution to the recurrence: $T(n)=8T(n/2)+cn$.

  (Answer: $\Theta(n^3)$.)

- Draw the recursion tree for this recurrence.

  a. What is its height?

  (Answer: $h=\log n$.)

  b. What is the number of leaves in the tree?

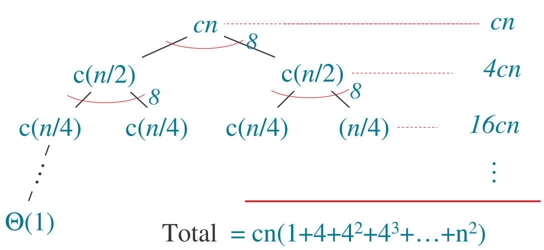
  (Answer: $8^h = 8^{\log n} = n^{\log 8} = n^3$.)

9/26/2007

S. Raskhodnikova; based on slides by E. Demaine, C. Leiserson, A. Smith, K. Wayne

Review questions: recursion tree

Solve $T(n)=8T(n/2)+cn$:



Total $= cn(1+4+4^2+4^3+\ldots+n^2)$

$= \Theta(n^3)$ *geometric series*

9/26/2007

S. Raskhodnikova; based on slides by E. Demaine, C. Leiserson, A. Smith, K. Wayne

Appendix: geometric series

$$1+x+x^2+\cdots+x^n = \frac{1-x^{n+1}}{1-x} \quad \text{for } x \neq 1$$

$$1+x+x^2+\cdots = \frac{1}{1-x} \quad \text{for } |x| < 1$$

Return to last slide viewed.

9/26/2007

S. Raskhodnikova; based on slides by E. Demaine, C. Leiserson, A. Smith, K. Wayne

Closest Pair of Points

Closest pair. Given n points in the plane, find a pair with smallest Euclidean distance between them.

Fundamental geometric primitive.
- Graphics, computer vision, geographic information systems, molecular modeling, air traffic control.
- Special case of nearest neighbor, Euclidean MST, Voronoi.

  fast closest pair inspired fast algorithms for these problems

Brute force. Check all pairs of points p and q with $\Theta(n^2)$ comparisons.

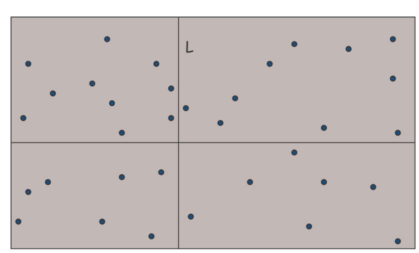
1-D version. O(n log n) easy if points are on a line.

Assumption. No two points have same x coordinate.

to make presentation cleaner

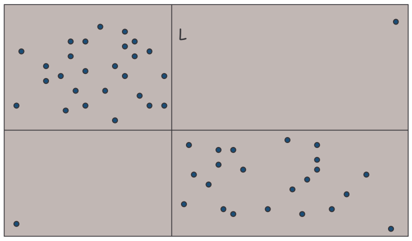
Closest Pair of Points: First Attempt

Divide. Sub-divide region into 4 quadrants.



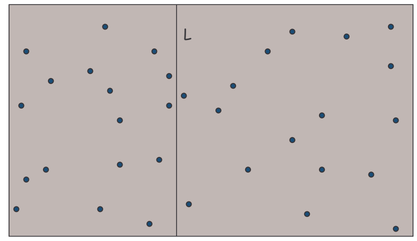
Closest Pair of Points: First Attempt

Divide. Sub-divide region into 4 quadrants.
Obstacle. Impossible to ensure n/4 points in each piece.



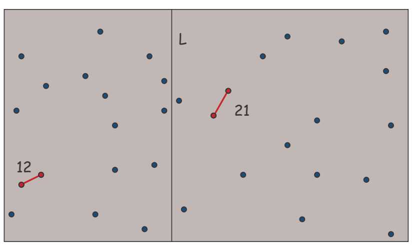
Closest Pair of Points

Algorithm.
- Divide: draw vertical line L so that roughly $\frac{1}{2}n$ points on each side.



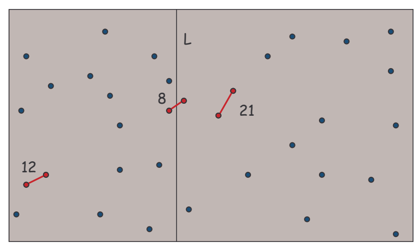
Closest Pair of Points

Algorithm.
- Divide: draw vertical line L so that roughly $\frac{1}{2}n$ points on each side.
- Conquer: find closest pair in each side recursively.



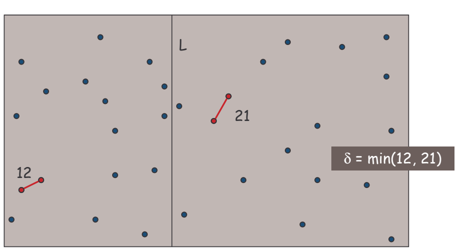
Closest Pair of Points

Algorithm.
- Divide: draw vertical line L so that roughly $\frac{1}{2}n$ points on each side.
- Conquer: find closest pair in each side recursively.
- Combine: find closest pair with one point in each side. ← seems like $\Theta(n^2)$
- Return best of 3 solutions.



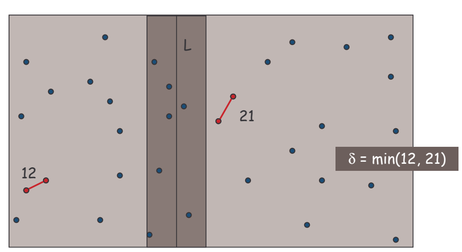
Closest Pair of Points

Find closest pair with one point in each side, assuming that distance < δ.



Closest Pair of Points

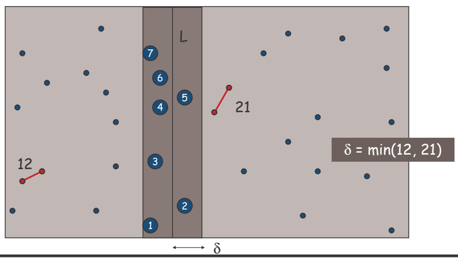
Find closest pair with one point in each side, assuming that distance < δ.
- Observation: only need to consider points within δ of line L.



Closest Pair of Points

Find closest pair with one point in each side, assuming that distance < δ.

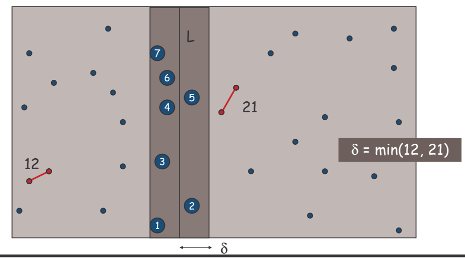
- Observation: only need to consider points within δ of line L.
- Sort points in 2δ-strip by their y coordinate.



Closest Pair of Points

Find closest pair with one point in each side, assuming that distance < δ.

- Observation: only need to consider points within δ of line L.
- Sort points in 2δ-strip by their y coordinate.
- Only check distances of those within 11 positions in sorted list!



Closest Pair of Points

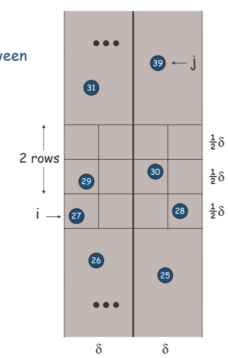
Def. Let s_i be the point in the 2δ-strip, with the i^{th} smallest y-coordinate.

Claim. If $|i - j| \ge 12$, then the distance between s_i and s_j is at least δ.

Pf.

- No two points lie in same $\frac{1}{2}\delta$-by-$\frac{1}{2}\delta$ box.
- Two points at least 2 rows apart have distance $\ge 2(\frac{1}{2}\delta)$. ▪

Fact. Still true if we replace 12 with 7.



Closest Pair Algorithm

```
Closest-Pair(p1, …, pn) {
   Compute separation line L such that half the points        O(n log n)
   are on one side and half on the other side.

   δ1 = Closest-Pair(left half)                                2T(n / 2)
   δ2 = Closest-Pair(right half)
   δ  = min(δ1, δ2)

   Delete all points further than δ from separation line L    O(n)

   Sort remaining points by y-coordinate.                     O(n log n)

   Scan points in y-order and compare distance between        O(n)
   each point and next 11 neighbors. If any of these
   distances is less than δ, update δ.

   return δ.
}
```

Closest Pair of Points: Analysis

Running time.

$$T(n) \le 2T(n/2) + O(n \log n) \Rightarrow T(n) = O(n \log^2 n)$$

Q. Can we achieve O(n log n)?

A. Yes. Don't sort points in strip from scratch each time.

- Each recursive returns two lists: all points sorted by y coordinate, and all points sorted by x coordinate.
- Sort by merging two pre-sorted lists.

$$T(n) \le 2T(n/2) + O(n) \Rightarrow T(n) = O(n \log n)$$