{ CS 323 | Lecture 11 }

DESIGN AND ANALYSIS & OF ALGORITHMS

Hoeteck Wee · hoeteck@cs.qc.cuny.edu

http://www.cs.qc.edu/~hoeteck/f09/

DIVIDE-AND-CONQUER ALGORITHMS

I. closest pair of points

II. integer multiplication

Closest pair of points

PROBLEM. given a list of n points in the plane $(x_1, y_1), \ldots, (x_n, y_n)$, find the pair that is closest.

APPLICATIONS. graphics, computer vision, molecular modeling, etc

NAIVE ALGORITHM. try all pairs of points, $O(n^2)$ time

TODAY. divide-and-conquer algorithm with running time $O(n \log n)$ time

- if $n \sim 10^3$, $O(n^2) \sim 10^6$ and $O(n \log n) \sim 10^4$
- use Manhattan distance $d((x,y),(x',y')) = |x - x'| + |y - y'|$

QUESTION. how to divide?

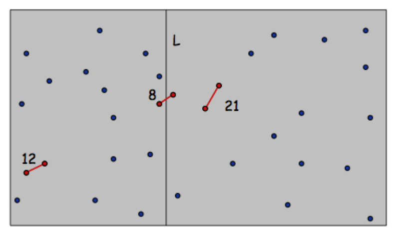
- expect $O(n \log n)$ time from $T(n) = 2T(n/2) + O(n)$
- need combining step to run in $O(n)$ time

Closest pair of points

ALGORITHM. divide-and-conquer

1. divide. draw a vertical line so that there are $\sim n/2$ points on each side
2. conquer. find closest pair on each side recursively
3. combine. find closest pair with one point on each side

return best of the 3 solutions



QUESTION. how to do divide?

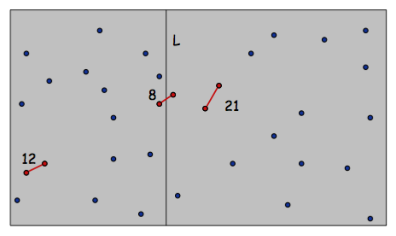
- sort points by x-coordinate: $O(n \log n)$ time
- provide sorted points as inputs to conquer step

Closest pair of points

ALGORITHM. divide-and-conquer

1. divide. draw a vertical line L so that there are $\sim n/2$ points on each side
2. conquer. find closest pair on each side recursively
3. combine. find closest pair with one point on each side

return best of the 3 solutions



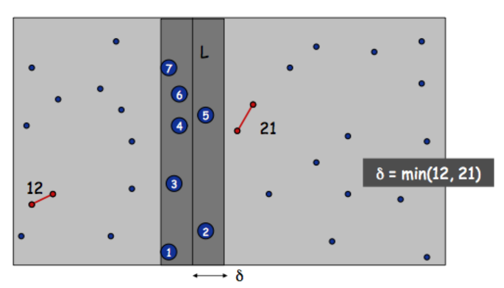
QUESTION. how to do combine in $O(n)$ time?

- easier: additionally assume distance $< \delta$
- use δ value from conquer step

Closest pair of points

QUESTION. how to do combine in $O(n)$ time?

- find closest pair with one point on each side, assuming distance $< \delta$



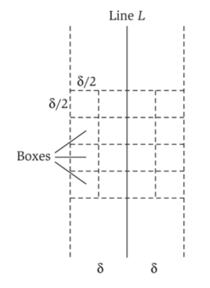
IDEAS. exploit distance $< \delta$

- observation 1: only need to consider a strip of width 2δ around L
- observation 2: compare each point with 15 (instead of $O(n)$) points
- figure out the 15 points by sorting points in strip by y coordinate

Closest pair of points

QUESTION. how to do combine in $O(n)$ time?

- find closest pair with one point on each side, assuming distance $< \delta$



each box contains ≤ 1 point

IDEAS. exploit distance $< \delta$

- observation 1: only need to consider a strip of width 2δ around L
- observation 2: compare each point with 15 (instead of $O(n)$) points
- figure out the 15 points by sorting points in strip by y coordinate

Integer multiplication

PROBLEM. multiply two n-bit numbers X and Y

ELEMENTARY APPROACH. $O(n^2)$ time

- $O(n)$ computation for each bit in Y
- n additions of $O(n)$-bit numbers

$$\begin{array}{r} 12 \\ \times\, 13 \\ \hline 36 \\ 12 \\ \hline 156 \end{array} \qquad \begin{array}{r} 1100 \\ \times\, 1101 \\ \hline 1100 \\ 0000 \\ 1100 \\ 1100 \\ \hline 10011100 \end{array}$$

ALGORITHM. divide-and-conquer

1. divide. write $X = 2^{n/2} \cdot A + B$ and $Y = 2^{n/2} \cdot C + D$
2. conquer. recursively compute $A \cdot C,\ A \cdot D,\ B \cdot C,\ B \cdot D$
3. combine. compute $2^n \cdot AC + 2^{n/2} \cdot (AD + BC) + BD$.

running time: $T(n) = 4T(n/2) + O(n) \ \Rightarrow\ T(n) = O(n^2)$

Integer multiplication

PROBLEM. multiply two n-bit numbers X and Y

ALGORITHM. divide-and-conquer

1. divide. write $X = 2^{n/2} \cdot A + B$ and $Y = 2^{n/2} \cdot C + D$
2. conquer. recursively compute $A \cdot C,\ A \cdot D,\ B \cdot C,\ B \cdot D$
3. combine. compute $2^n \cdot AC + 2^{n/2} \cdot (AD + BC) + BD$.

running time: $T(n) = 4T(n/2) + O(n) \;\Rightarrow\; T(n) = O(n^2)$

ALGORITHM. improved divide-and-conquer

- compute $AD + BC$ more quickly
- note $AD + BC = (A+B)(C+D) - AC - BD$

2. conquer. recursively compute $A \cdot C,\ B \cdot D,\ (A+B)(C+D)$

running time: $T(n) = 3T(n/2) + O(n) \;\Rightarrow\; T(n) = O(n^{\log_2 3}) \sim n^{1.59}$

THE END