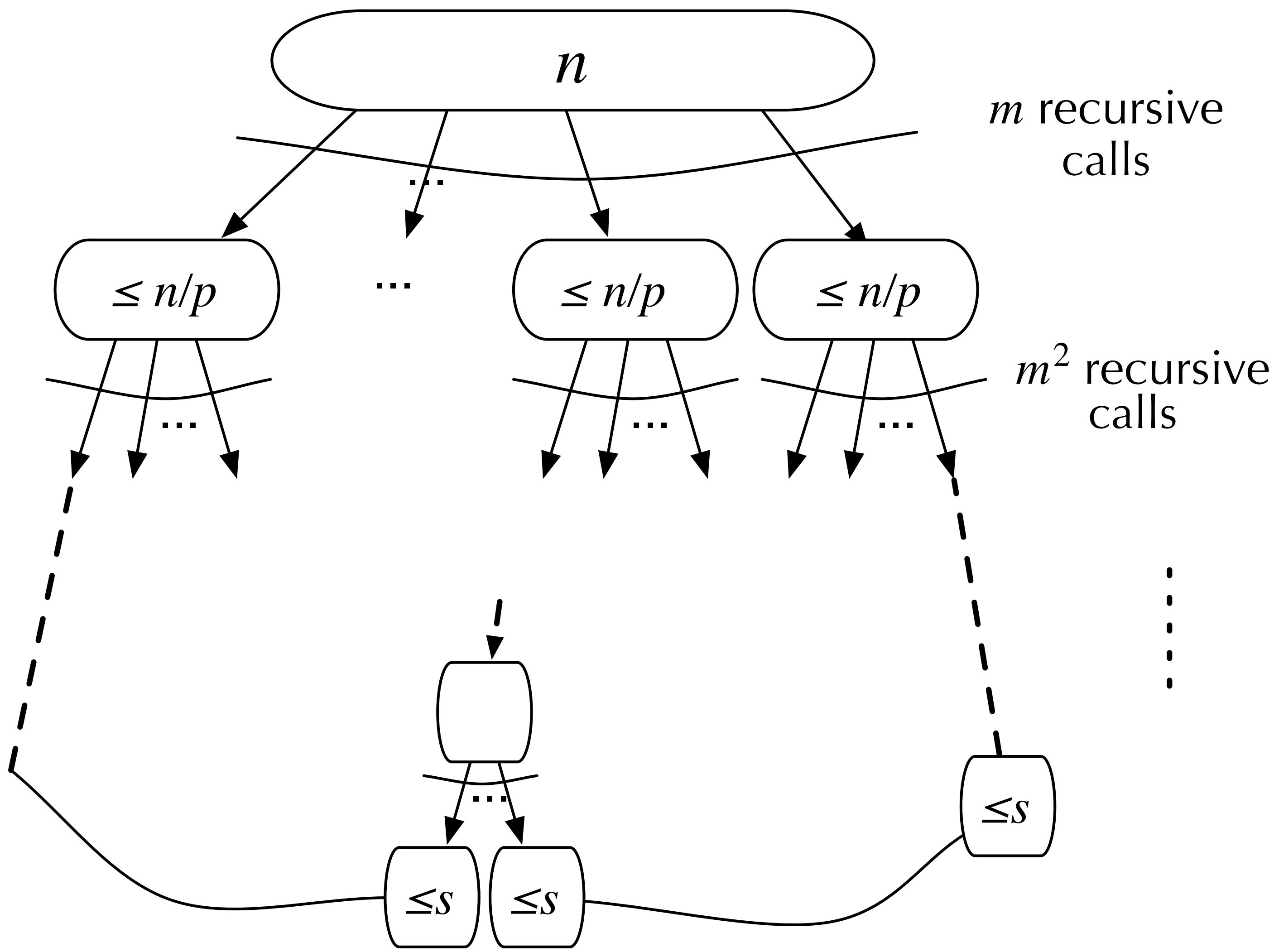
CSE202
Design and Analysis of Algorithms

Week 5 — Divide & Conquer 4: Master Theorem; Advanced Example

I. Master Theorem

Divide and Conquer



Complexity

$$C(n) \leq mC(\lceil n/p \rceil) + f(n)$$

for $n \geq p$

Aim:

$$C(n) = O(\text{simple}(n))$$

Warm Up: split in 2, 3 recursive calls, simple f, n a power of 2

Recall.

$$3^{\log_2 n} = n^{\log_2 3}.$$

$$C(n) \leq 3C(n/2) + cn^\alpha$$

iterate once

$$\leq cn^\alpha + 3c(n/2)^\alpha + 9C(n/4)$$

rearrange

$$\leq cn^\alpha(1 + 3/2^\alpha) + 9C(n/4)$$

iterate k-1 times

$$\leq cn^\alpha(1 + 3/2^\alpha + \cdots + (3/2^\alpha)^{k-1}) + 3^k C(n/2^k)$$

use $k = \log_2 n$

$$\leq cn^\alpha(1 + 3/2^\alpha + \cdots + (3/2^\alpha)^{k-1}) + O(3^k)$$

bound geometric series

$$\leq \underbrace{O(n^{\log_2 3})}_{\text{not larger}} + cn^\alpha \times \begin{cases} O(1), & \text{if } 2^\alpha > 3, \\ \log_2 n, & \text{if } 2^\alpha = 3, \\ O(n^{\log_2(3)-\alpha}), & \text{if } 2^\alpha < 3. \end{cases}$$

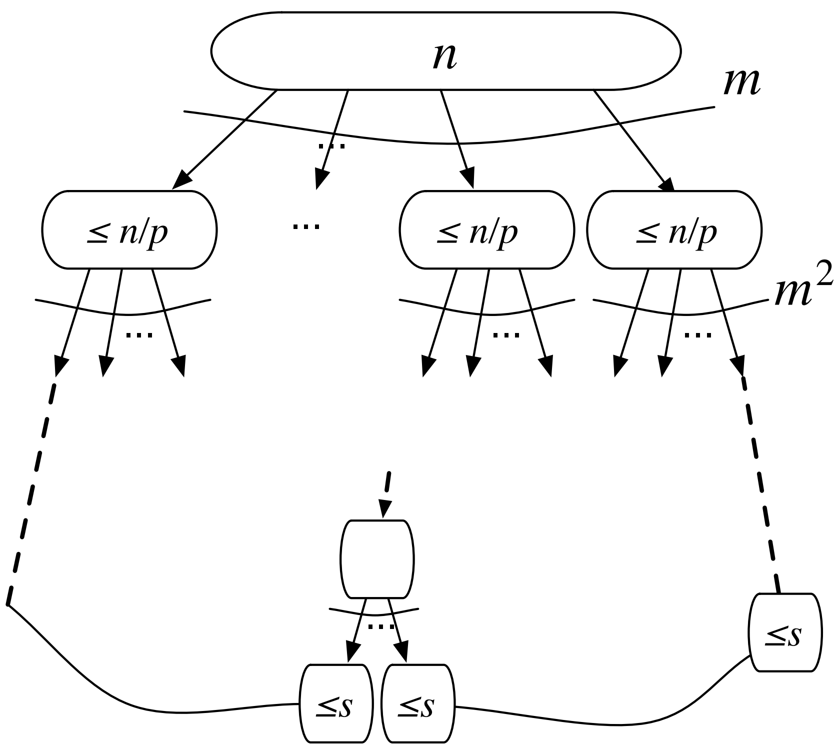
Master Theorem — Version 1

Assume $C(n) \le mC(\lceil n/p \rceil) + f(n)$ if $n \ge p$,

with $f(n) = cn^\alpha$ $(\alpha \ge 0)$. Let $q = p^\alpha$.

Then, as $n \to \infty$,

$$C(n) = \begin{cases} O(n^\alpha), & \text{if } q > m, \\ O(n^\alpha \log n), & \text{if } q = m, \\ O(n^{\log_p m}) & \text{if } q < m. \end{cases}$$



q/m governs which part of the recursion tree dominates

Examples

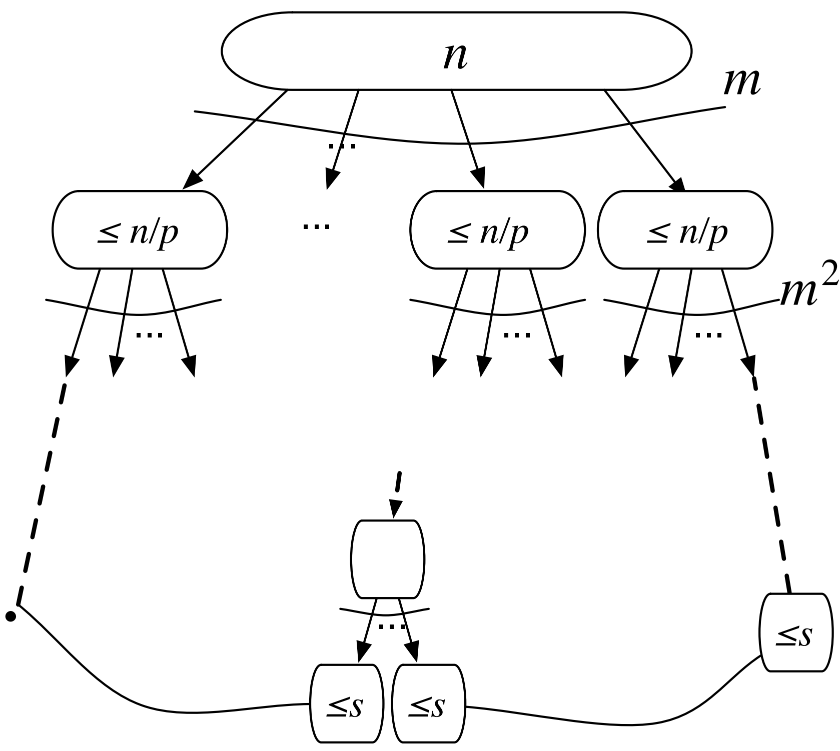
1. With $q = m$

Merge sort, FFT:

$$p = 2,\ m = 2,\ \alpha = 1,\ q = 2, \quad O(n \log n)\,.$$

Binary search, binary powering:

$$p = 2,\ m = 1,\ \alpha = 0,\ q = 1, \quad O(\log n)\,.$$



2. With $q < m$

Karatsuba:

$$p = 2,\ m = 3,\ \alpha = 1,\ q = 2, \quad O(n^{\log_2 3})\,.$$

Toom-Cook 3:

$$p = 3,\ m = 5,\ \alpha = 1,\ q = 3, \quad O(n^{\log_3 5})\,.$$

Strassen:

$$p = 2,\ m = 7,\ \alpha = 2,\ q = 4, \quad O(n^{\log_2 7})\,.$$

Proof when n is a Power of p

Same as before, for a general m and p

$$C(n) \le mC(n/p) + f(n)$$

iterate once

$$\le f(n) + mf(n/p) + m^2 C(n/p^2)$$

Recall.

$$m^{\log_p n} = n^{\log_p m}.$$

use hyp on f

$$\le f(n)(1 + m/q) + m^2 C(n/p^2)$$

iterate k-1 times

$$\le f(n)(1 + m/q + \cdots + (m/q)^{k-1}) + m^k C(n/p^k)$$

use $k = \log_p n$

$$\le f(n)(1 + m/q + \cdots + (m/q)^{k-1}) + O(m^k)$$

bound geometric series

$$\le \underbrace{O(n^{\log_p m})}_{\text{not larger}} + \underbrace{f(n)}_{cn^{\log_p q}} \times \begin{cases} O(1), & \text{if } q > m, \\ \log_p n, & \text{if } q = m, \\ O(n^{\log_p (m/q)}), & \text{if } q < m. \end{cases}$$

Proof in the General Case

Notation:

$$\lceil x/p \rceil_1 = \lceil x/p \rceil$$

$$\lceil x/p \rceil_{k+1} = \big\lceil \lceil x/p \rceil_k / p \big\rceil$$

N : power of p s.t. $n \le N < pn$

$$C(n) \le mC(\lceil n/p \rceil) + f(n)$$

iterate once

$$\le f(n) + mf(\lceil n/p \rceil) + m^2 C(\lceil n/p \rceil_2)$$

use N and f increasing

$$\le f(N) + mf(N/p) + m^2 C(\lceil n/p \rceil_2)$$

use hyp on f

$$\le f(N)(1 + m/q) + m^2 C(\lceil n/p \rceil_2)$$

iterate $\log_p N$ times, bound geom. series.

$$\le \underbrace{O(N^{\log_p m})}_{\text{not larger}} + f(N) \times \begin{cases} O(1), & \text{if } q > m, \\ \log_p N, & \text{if } q = m, \\ O(N^{\log_p (m/q)}), & \text{if } q < m\,. \end{cases}$$

use hyp on f

Conclude with $N < pn \Rightarrow f(N) \le p^{\alpha} f(n) = O(f(n))$.

Master Theorem — Even More General

Assume $C(n) \le mC(\lceil n/p \rceil) + f(n)$ if $n \ge p$, with $f(n)$ increasing and there exist (q, r) s.t. $q \le f(pn)/f(n) \le r$ for large enough n.

Then, as $n \to \infty$,

$$C(n) = \begin{cases} O(f(n)), & \text{if } q > m, \\ O(f(n) \log n), & \text{if } q = m, \\ O(f(n) n^{\log_p (m/q)}) & \text{if } q < m\,. \end{cases}$$

Note 1. When $f(n) = cn^{\alpha}$, then $q = r = p^{\alpha}$.

The previous result is a special case.

Exercise:
Treat the case
$f(n) = cn^{\alpha} \log^{\beta} n$.

Note 2. A tighter value of q gives a better complexity bound.

Examples

1. All the previous examples.

2. Newton's method for reciprocal, square root:

$$C(n) \le C(\lceil n/2 \rceil) + O(\mathrm{Mul}(n)),$$

and Mul satisfies

$$\begin{cases} \mathrm{Mul}(n_1) + \mathrm{Mul}(n_2) \le \mathrm{Mul}(n_1 + n_2) \\ \mathrm{Mul}(mn) \le m^2\,\mathrm{Mul}(n) \end{cases} \Longrightarrow 2 \le \frac{\mathrm{Mul}(2n)}{\mathrm{Mul}(n)} \le 4.$$

Conclusion: $q > m$,

$$p = 2,\ m = 1,\ q = 2,\ r = 4, \qquad C(n) = O(\mathrm{Mul}(n))\,.$$

Proof when n is a Power of p

Copy-pasted from before

$$C(n) \le mC(n/p) + f(n)$$

iterate once

$$\le f(n) + mf(n/p) + m^2C(n/p^2)$$

use hyp on f

$$\le f(n)(1 + m/q) + m^2C(n/p^2)$$

iterate k-1 times

$$\le f(n)(1 + m/q + \cdots + (m/q)^{k-1}) + m^kC(n/p^k)$$

use $k = \log_p n$

$$\le f(n)(1 + m/q + \cdots + (m/q)^{k-1}) + O(m^k)$$

bound geometric series

$$\le \underbrace{O(n^{\log_p m})}_{\text{not larger}} + f(n) \times \begin{cases} O(1), & \text{if } q > m, \\ \log_p n, & \text{if } q = m, \\ O(n^{\log_p (m/q)}), & \text{if } q < m\,. \end{cases}$$

Lemma.

$$n^{\log_p q} = O(f(n))\,.$$

Proof on the blackboard

was easy before

Proof in the General Case

Copy-pasted from before

$$C(n) \leq mC(\lceil n/p \rceil) + f(n)$$

iterate once

$$\leq f(n) + mf(\lceil n/p \rceil) + m^2 C(\lceil n/p \rceil_2)$$

use N and f increasing

$$\leq f(N) + mf(N/p) + m^2 C(\lceil n/p \rceil_2)$$

use hyp on f

$$\leq f(N)(1 + m/q) + m^2 C(\lceil n/p \rceil_2)$$

iterate $\log_p N$ times, bound geom. series.

$$\leq \underbrace{O(N^{\log_p m})}_{\text{not larger}} + f(N) \times \begin{cases} O(1), & \text{if } q > m, \\ \log_p N, & \text{if } q = m, \\ O(N^{\log_p (m/q)}), & \text{if } q < m. \end{cases}$$

use hyp on f

Conclude with $N < pn \Rightarrow f(N) \leq rf(n) = O(f(n))$.

was p^α before

Notation:

$\lceil x/p \rceil_1 = \lceil x/p \rceil$

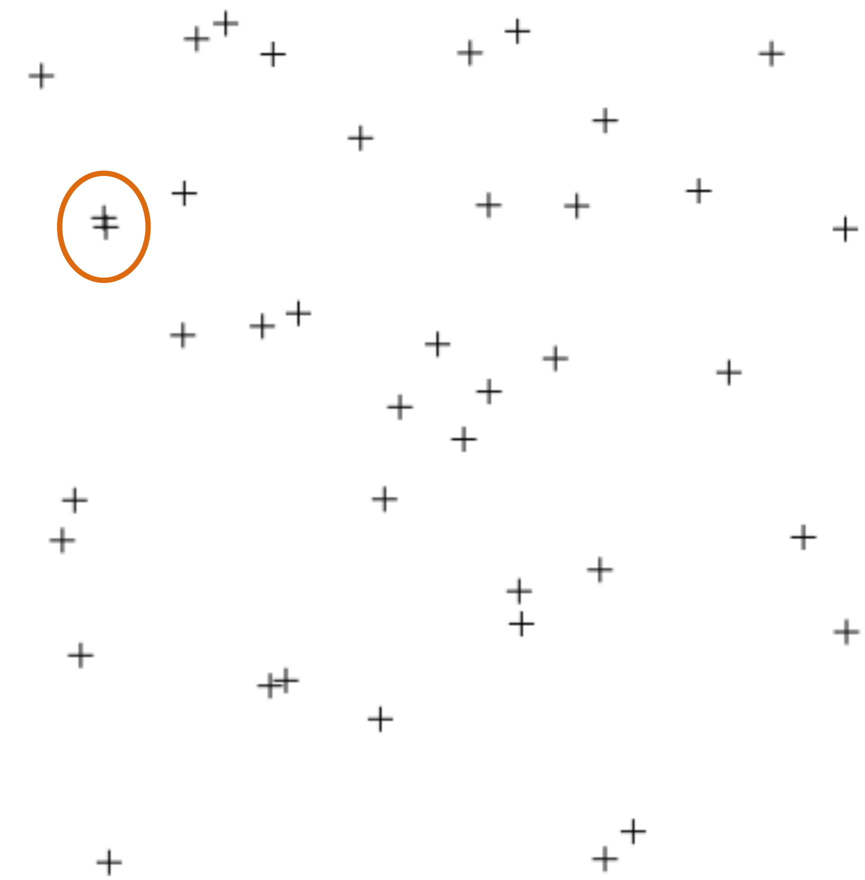
$\lceil x/p \rceil_{k+1} = \lceil \lceil x/p \rceil_k / p \rceil$

N: power of p s.t. $n \leq N < pn$

II. Closest Pair of Points

a case when merging sub-results is not so easy

Statement of the Problem

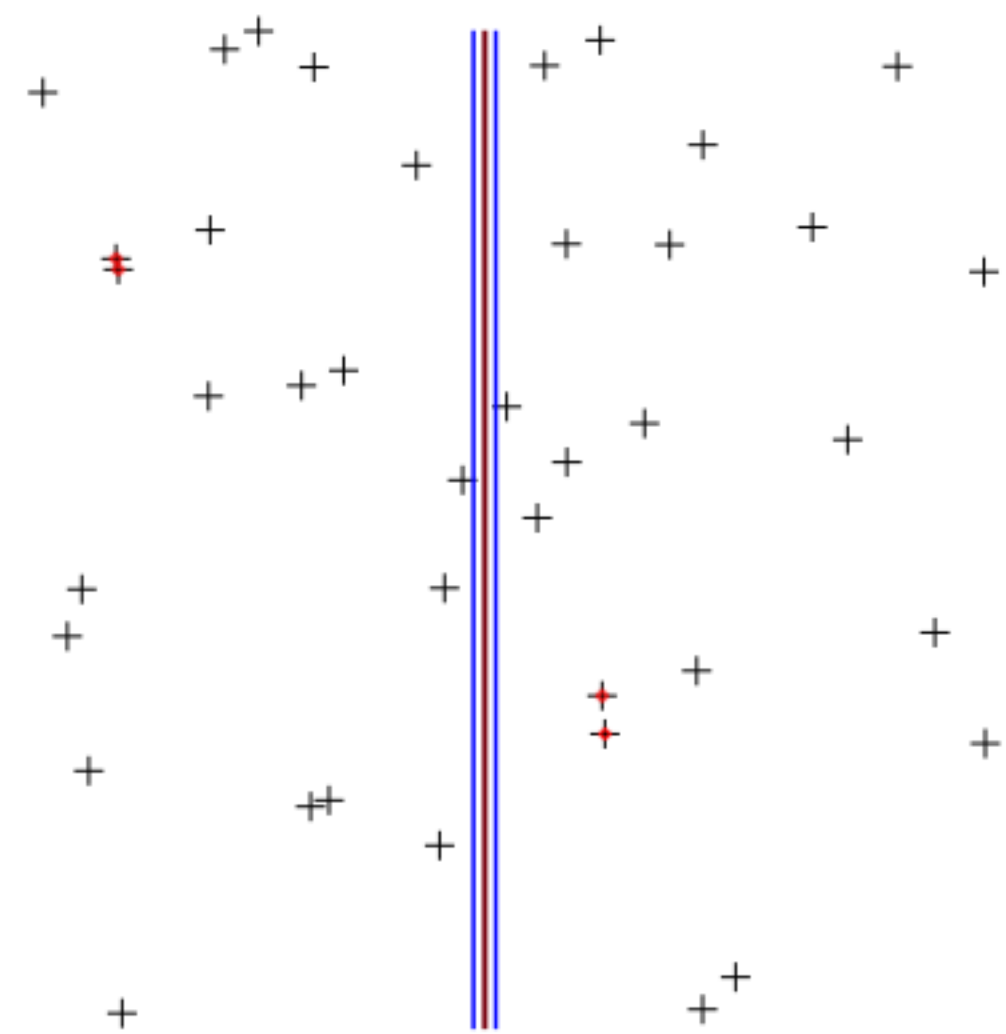


Given n points
in the plane,
find the closest pair.

Naive method: compute all $O(n^2)$ pairwise distances,
return pair with the smallest one.

Divide and Conquer: split points into left and right,
solve both subproblems (ok), recombine (hard).

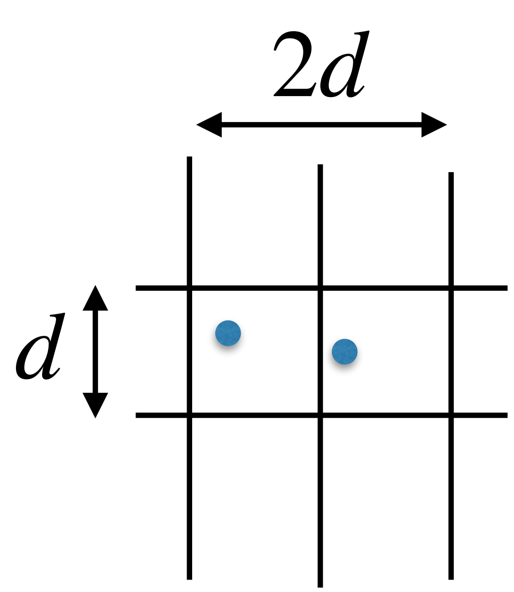
Divide and Conquer



Sort the points by x-coordinate and cut in the middle

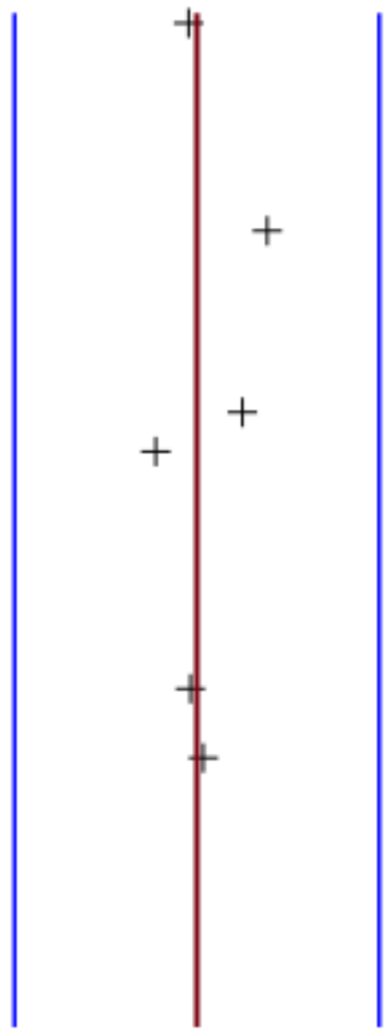
Time: $O(n \log n)$

Naive recombination: $O(n^2)$ pairs (left,right)

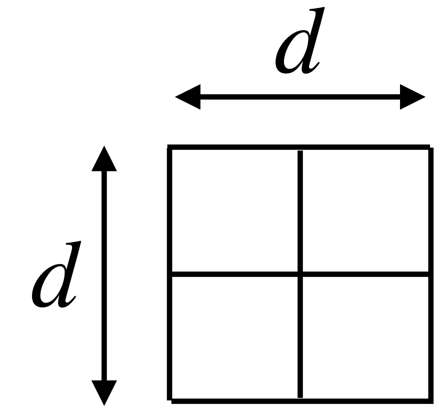


1st observation: if d is the minimal distance on both sides, then it is sufficient to focus on a strip of width $2d$ around the middle.

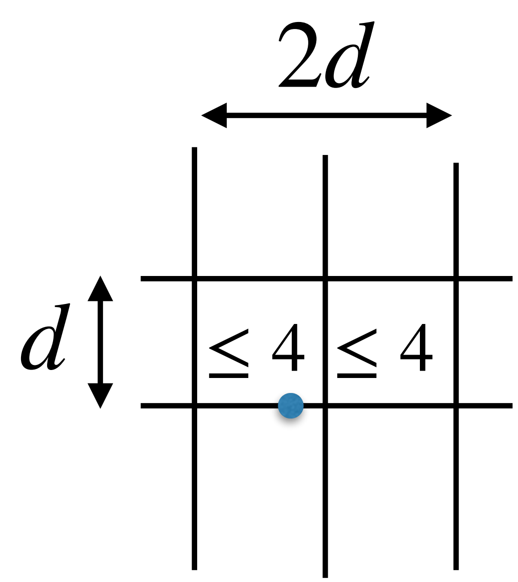
Comparisons within a Strip



2nd observation: At most 4 points at distance $\geq d$ from one another can lie in a $d \times d$ square.



≤1 in each sub-square



Conclusion. Each point has to be compared with at most 7 of the next ones for the *y*-coordinate.

Algorithm ClosestPair

Input: P: array of pairs of coordinates

X,Y:=indices(P) sorted by x, y- coordinate

Output: $\min_{i \neq j} d(P[i], P[j])$

Complexity

$$C(n) \leq 2C(\lceil n/2 \rceil) + O(n)$$
$$= O(n \log n) .$$

was the difficult part

base cases

if $|X| = 1$ return ∞

if $|X| = 2$ return $d(P[X[1]], P[X[2]])$

divide

$k = \lceil |X|/2 \rceil; X_m = P[X[k]] \,.\, x$

$(X_\ell, X_r) =$ split X at index k

$(Y_\ell, Y_r) =$ split Y depending on $\operatorname{sgn}(P[Y[i]] \,.\, x - X_m)$

recurse

$d_\ell = \text{ClosestPair}(P, X_\ell, Y_\ell);\;\; d_r = \text{ClosestPair}(P, X_r, Y_r)$

recombine

$d = \min(d_\ell, d_r); \quad U = [i \in Y \mid P[i] \,.\, x \in [X_m - d, X_m + d]]$

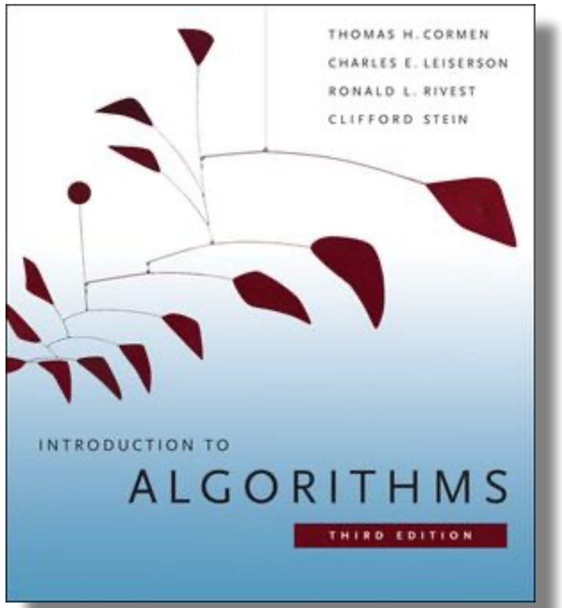
$d' = \min\{d(P[U[i]], P[U[j]]) \mid i < |U|, i < j \leq \min(i + 7, |U|)\}$

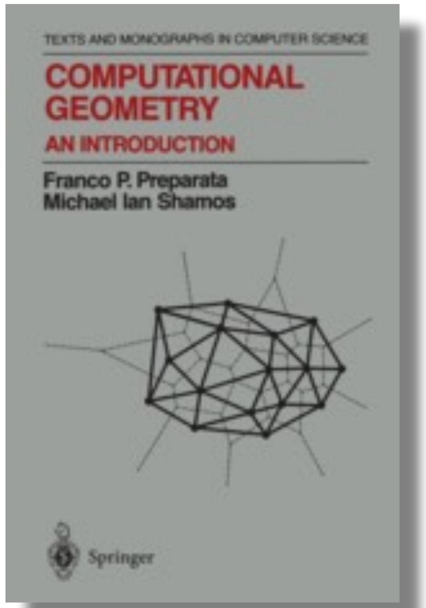
return $\min(d, d')$

References for this lecture

The slides are designed to be self-contained.

They were prepared using the following books that I recommend if you want to learn more:





Next

Assignment this week: optimized divide-and-conquer

Next tutorial: DAC + use of the Master Theorem

Next week: Randomization 1.

Feedback

Moodle for the slides, TDs and exercises.

Questions or comments: Bruno.Salvy@inria.fr