CS 341: Algorithms

Lecture 4: Divide and conquer, continued

Éric Schost

based on lecture notes by many other CS341 instructors

David R. Cheriton School of Computer Science, University of Waterloo

Fall 2024

Closest pairs

Closest pairs

Goal: given n points (x_i, y_i) in the plane, find a pair (i, j) that minimizes the distance

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Equivalent to minimize

$$d_{i,j}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$$

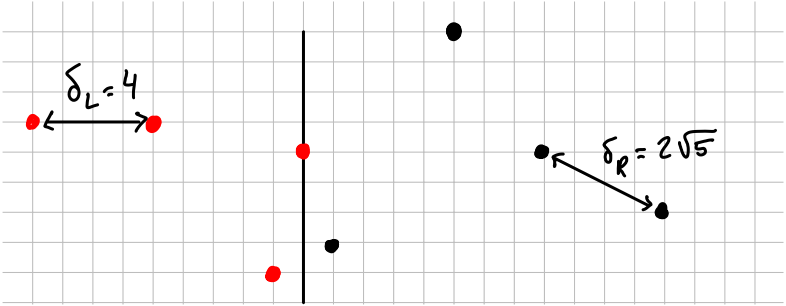
Assumption

all x_i's distinct

Divide-and-conquer

Idea: separate the points into two halves L, R at the median x-value

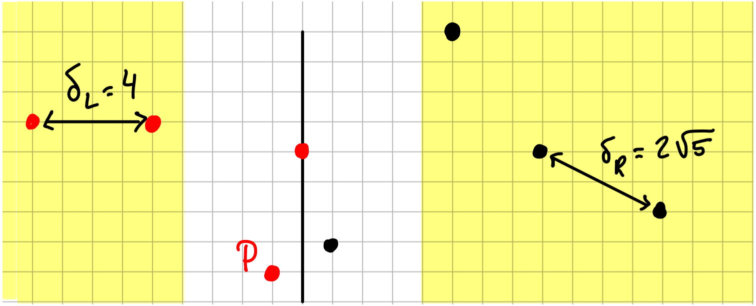
- $L =$ all $n/2$ points with $x \leq x_{\text{median}}$
- $R =$ all $n/2$ points with $x > x_{\text{median}}$
- find the closest pairs in both L and R recursively
- the closest pair is either **between points in L** (done), or **between points in R** (done), or **transverse** (one in L, one in R)



Finding the shortest transverse distance

Set $\delta = \min(\delta_L, \delta_R)$

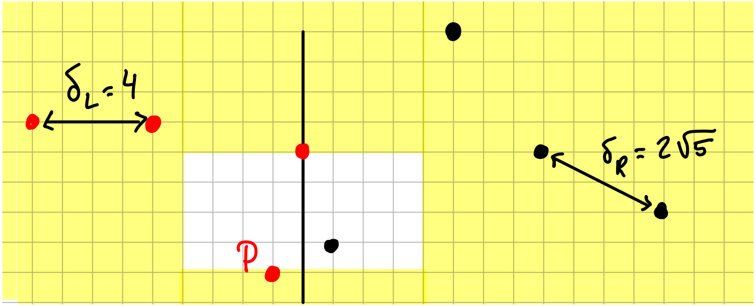
- We only need to consider transverse pairs (P, Q) with $\mathbf{dist}(\boldsymbol{P}, \boldsymbol{R}) \leq \boldsymbol{\delta}$ and $\mathbf{dist}(\boldsymbol{Q}, \boldsymbol{L}) \leq \boldsymbol{\delta}$.



Finding the shortest transverse distance

Set $\delta = \min(\delta_L, \delta_R)$

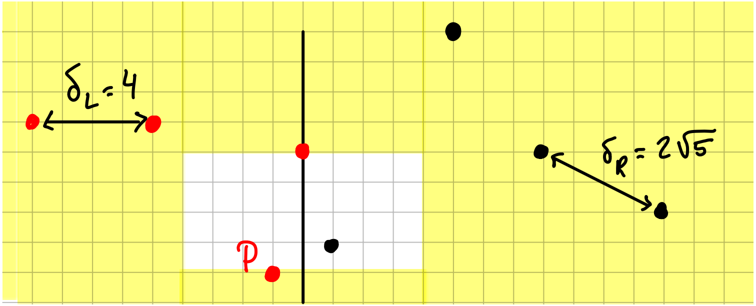
- For any $P = (x_P, y_P)$, enough to look at points with $y_P \leq y < y_P + \delta$



Finding the shortest transverse distance

Set $\delta = \min(\delta_L, \delta_R)$

- For any $P = (x_P, y_P)$, enough to look at points with $y_P \leq y < y_P + \delta$



So it is enough to check distances $d(P, Q)$ for Q in the rectangle.

How many points in the rectangle?

Claim

There are at most **8** points from our initial set (including P) in the rectangle.

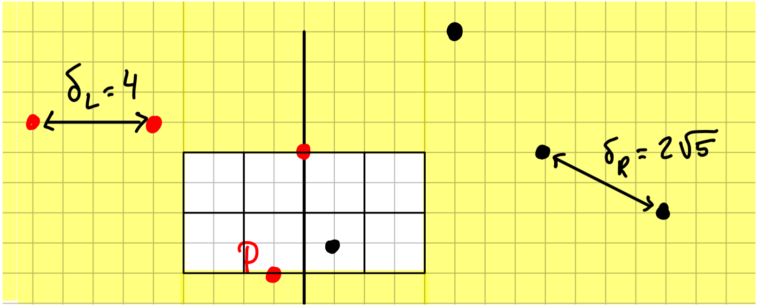
How many points in the rectangle?

Claim

There are at most **8** points from our initial set (including P) in the rectangle.

Proof. Cover the rectangle with **8** squares of side length $\delta/2$

- they overlap along lines, but it's OK



How many points in the rectangle?

Claim

There are at most **8** points from our initial set (including P) in the rectangle.

Proof. Cover the rectangle with **8** squares of side length $\delta/2$

- they overlap along lines, but it's OK
- a square on the left contains **at most one point from** L
- a square on the right contains **at most one point from** R

Consequence: at most 8 points in the range $y_P \leq y \leq y_P + \delta$

Data structures and runtime

Initialization: sort the points **twice**, with respect to x and to y.
One-time cost $O(n\log(n))$, before recursive calls — cf kd-trees

Recursive calls. Enter with two lists (points sorted in x and in y).

- finding the x-median is easy — $\Theta(1)$
- for the next recursive calls, split the sorted lists — $\Theta(n)$
- remove the points at distance $\geq \delta$ from the x-splitting line — $\Theta(n)$
- inspect all remaining points P in increasing y-order.

  For each P, compute the distance to the points with $y_P \leq y \leq y_P + \delta$ and keep the min. At most 8 points per P. — $\Theta(n)$

Runtime: $T(n) = 2T(n/2) + cn$ so $T(n) \in \Theta(n\log(n))$

Linear time median

Beyond the master theorem: median of medians

Median: given $A[0..n-1]$, find the entry that would be at index $\lfloor n/2 \rfloor$ **if A was sorted**

Selection: given $A[0..n-1]$ and k in $\{0, \ldots, n-1\}$, find the entry that would be at index k **if A was sorted**

Known results: sorting A in $\boldsymbol{\Theta(n \log(n))}$, or a simple randomized algorithm in expected time $\boldsymbol{\Theta(n)}$

Assumption

all $A[i]$'s distinct

The selection algorithm

```
quick-select(A, k)
A: array of size n,   k: integer s.t. 0 ≤ k < n
    p ← choose-pivot(A)
    i ← partition(A, p)                         i is the correct index of p
    if i = k then
        return A[i]
    else if i > k then
        return quick-select(A[0, 1, ..., i − 1], k)
    else if i < k then
        return quick-select(A[i + 1, i + 2, ..., n − 1], k − i − 1)
```

partition(A, p)**:**

- reorders A so that $A = [\leq \boldsymbol{p},\ \ \boldsymbol{A[i] = p},\ \ \geq \boldsymbol{p}]$
- if all entries distinct, $A = [< \boldsymbol{p},\ \ \boldsymbol{A[i] = p},\ \ > \boldsymbol{p}]$

Goal: find a pivot such that both $\boldsymbol{i}$ and $\boldsymbol{n - i - 1}$ are not too large

Median of medians

Sketch of the algorithm:

- divide A into $n/5$ groups $G_{\mathbf{1}}, \ldots, G_{\boldsymbol{n/5}}$ of size 5
- find the medians $m_{\mathbf{1}}, \ldots, m_{\boldsymbol{n/5}}$ of each group $\boldsymbol{\Theta(n)}$
- pivot p is the median of $[m_{\mathbf{1}}, \ldots, m_{\boldsymbol{n/5}}]$ $\boldsymbol{T(n/5)}$

Claim

With this choice of p, the indices $\boldsymbol{i}$ and $\boldsymbol{n-i-1}$ are **at most $\boldsymbol{7n/10}$**

Median of medians

Sketch of the algorithm:

- divide A into $n/5$ groups $G_{\mathbf{1}}, \ldots, G_{\boldsymbol{n/5}}$ of size 5
- find the medians $m_{\mathbf{1}}, \ldots, m_{\boldsymbol{n/5}}$ of each group $\boldsymbol{\Theta(n)}$
- pivot p is the median of $[m_{\mathbf{1}}, \ldots, m_{\boldsymbol{n/5}}]$ $\boldsymbol{T(n/5)}$

Claim

With this choice of p, the indices $\boldsymbol{i}$ and $\boldsymbol{n-i-1}$ are **at most** $\boldsymbol{7n/10}$

Proof

- **half** of the m_i's are greater than or equal to p $\boldsymbol{n/10}$
- for each m_i, there are **3** elements in G_i greater than or equal to m_i
- so **at least** $\boldsymbol{3n/10}$ elements greater than p
- so **at most** $\boldsymbol{7n/10}$ elements less than p
- so i is **at most** $\boldsymbol{7n/10}$. Same thing for $n-i-1$

Median of medians

Sketch of the algorithm:

- divide A into $n/5$ groups $G_{\mathbf{1}}, \ldots, G_{\boldsymbol{n/5}}$ of size 5
- find the medians $m_{\mathbf{1}}, \ldots, m_{\boldsymbol{n/5}}$ of each group $\boldsymbol{\Theta(n)}$
- pivot p is the median of $[m_{\mathbf{1}}, \ldots, m_{\boldsymbol{n/5}}]$ $\boldsymbol{T(n/5)}$

Claim

With this choice of p, the indices $\boldsymbol{i}$ and $\boldsymbol{n-i-1}$ are **at most $\boldsymbol{7n/10}$**

Consequence: (sloppy) recurrence

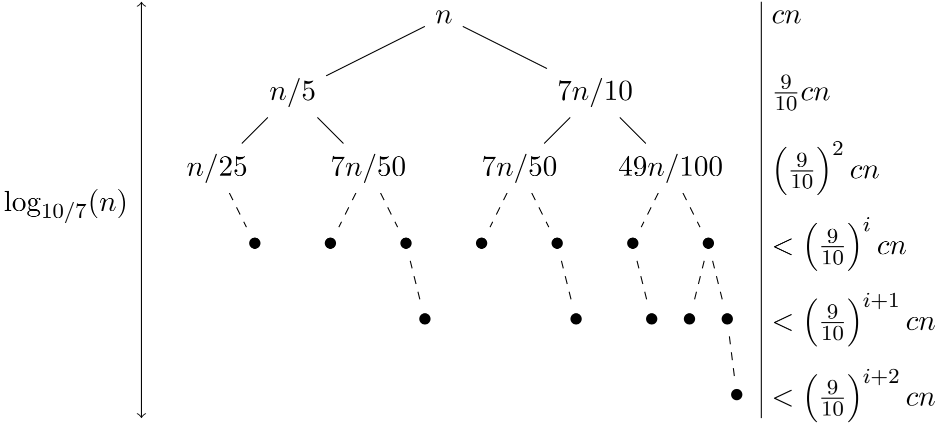
$$T(n) = T(n/5) + T(7n/10) + cn$$

Claim

This gives worst-case $\boldsymbol{T^w(n) \in \Theta(n)}$ ($\Omega(n)$ clear)

The recursion tree for $T(n)$

Can prove: enough to analyze the sloppy recurrence, setting $T(n) = 0$ for $n \leq 1$.



Geometric sum of ratio $\mathbf{9/10 < 1}$, so $\boldsymbol{\Theta(n)}$.

Final remarks

1. Why not median of three?

- we do $n/3$ **groups of 3** and find their medians $m_1, \ldots, m_{n/3}$ $\Theta(n)$
- p is the median of $[m_1, \ldots, m_{n/3}]$ $T(n/3)$
- half of the m_i's are greater than or equal to p $n/6$
- in each group, 2 elements greater than or equal to m_i
- so overall at least $n/3$ elements greater than or equal to p
- so at most $2n/3$ elements less than p
- so $i \leq 2n/3$, and $n - 1 - i \leq 2n/3$

Recurrence: $T(n) = T(n/3) + T(2n/3) + cn$

Final remarks

1. Why not median of three?

- we do $n/3$ **groups of** 3 and find their medians $m_1, \ldots, m_{n/3}$ $\Theta(n)$
- p is the median of $[m_1, \ldots, m_{n/3}]$ $T(n/3)$
- half of the m_i's are greater than or equal to p $n/6$
- in each group, 2 elements greater than or equal to m_i
- so overall at least $n/3$ elements greater than or equal to p
- so at most $2n/3$ elements less than p
- so $i \leq 2n/3$, and $n - 1 - i \leq 2n/3$

Recurrence: $T(n) = T(n/3) + T(2n/3) + cn$

2. Handling duplicates

- **option 1.** revisit partition: $[< p, \quad p, \ldots, p, \quad > p]$
- **option 2.** break ties: $A[i] \to [A[i], i]$