Michael A. Forbes
MiForbes@illinois.edu
2024-01-18.1

CS473 Algorithms : Lecture 2 (2023-01-18)

logistics : - pset 0 out F17
- piazza [read course website]
- signup - gradescope
[logistics]

unbox page

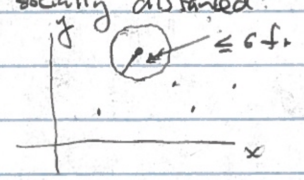
last lecture : - introduction [motivation and goals]
- divide and conquer : integer multiplication : Karatsuba's algo

today : - divide and conquer
[move to front]
[laptop policy]
[covid]

[move to front]

Q : are we socially distanced?

[closest pair]
[computational geometry]



def : given points $P = \{p_1, \ldots, p_n\}$ $p_i = (x_i, y_i) \in \mathbb{Z}^2$, the closest pair problem is to find $\underset{\text{arg}}{\hat{\ }} \min_{i \neq j} \text{dist}(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ [euclidean dist]

[core of problem]

[value]
[finding minimizer]
[usually two are comparable]

$\boxed{\text{assuming}}$ all x_i, y_j distinct [methods]
[address foundations]

Q : cost of manipulating integers?
[possible ans]

A : programming lang have $a \leftarrow b \cdot c$ as primitive op [unit cost]
[last lecture] [conflict]

A : multiplication of n-bit integers in $O(n^{\log_2 3})$ steps [is n large? or small?]

Q : what is a $\boxed{\text{reasonable}}$ cost model of manipulating integers?
[possible ans] [models "cheap" as unit cost]

A : n-bit arithmetic at unit cost
[but]
fact : exist algorithms "efficient" in this model w/ no known efficient $\boxed{\text{real-life}}$ implementation → not realistic
[only bit ops are primitive]

A : n-bit arithmetic in $n^{O(1)}$ time → tedious
[realistic]

A : $O(\lg n)$-bit arithmetic ops at unit cost ⇒ compromise [simpler analysis]
[unless otherwise specified]

convention : on problems w/ a set of n integer inputs, the integers are $O(\lg n)$ bits
algorithms can use $O(\lg n)$-bit arithmetic at unit cost
[runtime measured as function of n]
the $\underline{\text{input size}}$ is n

eg : closest pair

eg/not : n-bit multiplication [2 numbers, but not O(1) size] [view numbers as sequence of bits]

⇐ [Q]
[closest pair]

prop : closest pair in $O(n^2)$ steps

pf : algo : output $\underset{\text{arg}}{\hat{\ }} \min_{i \neq j} \text{dist}(p_i, p_j)^2 = (x_i - x_j)^2 + (y_i - y_j)^2$ [omit sqrt, so $\mathbb{Z}$]

correctness : $\sqrt{\cdot}$ is monotonic

complexity : $\binom{n}{2} = O(n^2)$ $O(1)$ cost, as $O(\lg n)$ bit arithmetic

∎

Michael A. Forbes
mforbes@illinois.edu
2024-01-18.2

Q: do better?

prop: one-dimensional closest pair in $O(n \lg n)$

x_i distinct

pf: algo: - sort $x_1, \ldots, x_n$ into $\hat{x}_1 \le \cdots \le \hat{x}_n$ — $O(n \lg n)$

- output $\min_i \mathrm{dist}(\hat{x}_i, \hat{x}_{i+1}) = \sqrt{(x_i - x_{i+1})^2} = |x_i - x_{i+1}| = x_{i+1} - x_i$ [x_i sorted] — $O(1)$

correctness: clear

complexity: sorting is $O(n \lg n)$ [cs374]

additional $O(n)$ ∎

$d \geq 2$

thm: $\boxed{\text{two-dimensional}}$ closest pair in $O(n \lg n)$ [$\ll n^2$]

x coord & y coord

idea: sort $p_1, \ldots, p_n$ by x-coord

$\boxed{\text{but}}$ ⋮ [closest point] [but $\boxed{\text{far}}$ in x-coord sorted order]

idea: divide and conquer

notation

def: $A, B \subseteq P$. $\mathrm{dist}(A,B) = \min_{\substack{a \neq b \\ a \in A,\ b \in B}} \mathrm{dist}(a,b)$

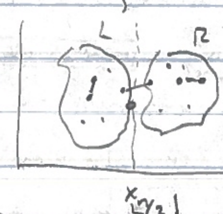
$\Rightarrow \mathrm{dist}(P,P)$ is closest pair

$\frac{n}{2}$ vs $\lfloor \frac{n}{2} \rfloor$

def: define $L, R \subseteq P$ by $L = \{p_i : x_i \le \hat{x}_{\lfloor n/2 \rfloor}\}$ [left] — median x-coord

$R = \{ \text{—} > \text{—} \}$ [right]

lem: $|L| = \lfloor n/2 \rfloor$, $|R| = \lceil n/2 \rceil$



[divide]
[conquer]

pf: all x-coord distinct [assumption] [≤ 1 point on this line]

framework

lem: $\mathrm{dist}(P,P) = \min \begin{cases} \mathrm{dist}(L,L) & \text{[recurse]} \\ \mathrm{dist}(R,R) & \text{[recurse]} \\ \mathrm{dist}(L,R) & \text{[combine]} \end{cases}$ [global min]

combine vs conquer

[proof clear]

Q: computing $\mathrm{dist}(L,R)$ still hard? [$\boxed{\text{not}}$ recursion]

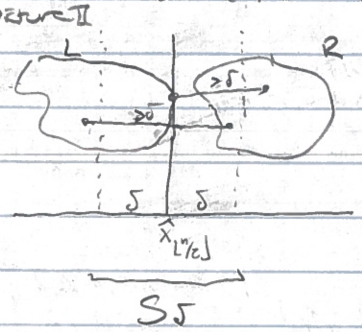
def: $S_\delta = \{p_i : \hat{x}_{\lfloor n/2 \rfloor} - \delta \le x_i \le \hat{x}_{\lfloor n/2 \rfloor} + \delta\}$, the δ-margin median strip of P

lem: $\delta := \min(\mathrm{dist}(L,L), \mathrm{dist}(R,R))$

then $\mathrm{dist}(P,P) = \min \begin{cases} \delta \\ \mathrm{dist}(L \cap S_\delta, R \cap S_\delta) \end{cases}$

pf: [picture]

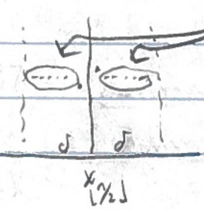
L touches $\hat{x}_{\lfloor n/2 \rfloor}$



[excludes $\mathrm{dist}(L, R \cap S_\delta)$]
[excludes $\mathrm{dist}(L \cap S_\delta, R)$]

Q: is computing $\mathrm{dist}(L \cap S_\delta, R \cap S_\delta)$ easier? [previous hard example]

eg: sort by y-coord



violates $\delta \le \mathrm{dist}(L,L)$, $\mathrm{dist}(R,R)$

Michael A. Forbes
miforbes@illinois.edu
2024-01-18.3

idea : $\boxed{\text{use}}$ $\delta \le dist(L,L), dist(R,R)$ in computing $dist(L\cap S_\delta, R\cap S_\delta)$ — lecture flow

prop [key] : $\delta = \min \begin{cases} dist(L,L) \\ dist(R,R) \end{cases}$

$S_\delta = (\tilde{p}_i)_i$ w/ $\tilde{p}_i$ sorted by y-coord [no need for other p_i] — $\sim$ vs $\wedge$

[defer proof] if $\tilde{p}_i \in L\cap S_\delta$, $\tilde{p}_j \in R_j \cap S_\delta$ w/ $dist(\tilde{p}_i, \tilde{p}_j) \le \delta \Rightarrow |i-j| \le 8$ [close, in sorted order] [constant!] [not exactly what was promised]

thm : two dimensional closest pair in $O(n(\lg n)^2)$ time

pf : **algo** :
- if $|P| \le 3$, solve directly [base case] [use quadratic algo] — $O(1)$
- sort P by x-coord — $O(n\lg n)$
- partition P by $\hat{x}_{\lfloor n/2 \rfloor}$ into $L \sqcup R$ — $O(n)$
- recursively compute — $dist(L,L)$ — $T(n/2)$ — flow
  — $dist(R,R)$ — $T(n/2)$
- $\delta = \min\{\ \}$ — $O(1)$
- compute S_δ — $O(n)$
- sort S_δ by y-coord — $O(n\lg n)$
- compute $\displaystyle\min_{\substack{\tilde{p}_i \in L\cap S_\delta \\ \tilde{p}_j \in R\cap S_\delta \\ |i-j| \le 8}} dist(\tilde{p}_i, \tilde{p}_j)$ — $O(8\cdot n) = O(n)$
- output $\min\{\delta, \ \}$ [modulo prop] — $O(1)$

correctness : clear

complexity : $T(n)$ = max #steps on n points

[accounting]

$T(n) \le 2\cdot T(n/2) + O(n\lg n) \le \cdots \le O(n(\lg n)^2)$ [solve] ∎

prop [key] : $\delta = \min \begin{cases} dist(L,L) \\ dist(R,R) \end{cases}$

$S_\delta = (\tilde{p}_i)_i$ w/ $\tilde{p}_i$ sorted by y-coord

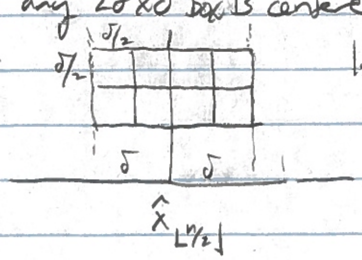
if $\tilde{p}_i \in L\cap S_\delta$, $\tilde{p}_j \in R\cap S_\delta$ w/ $dist(\tilde{p}_i, \tilde{p}_j) \le \delta \Rightarrow |i-j| \le 8$

pf : **clm** : any $\delta/2 \times \delta/2$ box contains ≤ 1 point from L
" R

pf : $\delta/2$ box with p, q : $dist(p,q) \le \sqrt{(\underbrace{x - z}_{|\cdot| \le \delta/2})^2 + (\underbrace{y-w}_{|\cdot|\le\delta/2})^2} \le \sqrt{\delta^2/4 + \delta^2/4} = \frac{\delta}{\sqrt{2}} < \delta \le \begin{cases} dist(L,L) \\ dist(R,R) \end{cases}$ [p] [q] [contrapositive] ∎

clm : any $2\delta \times \delta$ box B centered around $\hat{x}_{\lfloor n/2 \rfloor}$ contains ≤ 8 points from S_δ

pf :



$|S_\delta \cap B| = \underbrace{|S_\delta \cap B \cap L|}_{\le 4 \text{ [clm]}} + \underbrace{|S_\delta \cap B \cap R|}_{\le 4}$ [L, R disjoint]

≤ 8

Michael A. Forbes
miforbes@illinois.edu
2024-01-18.4

pf of prop:

$\tilde{p}_i = (\tilde{x}_i, \tilde{y}_i) \in L \cap S_\delta$

$\tilde{p}_j = (\tilde{x}_j, \tilde{y}_j) \in R \cap S_\delta$

w/ - $\tilde{y}_i \le \tilde{y}_j$

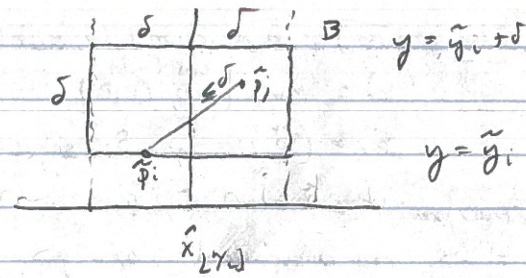
- $dist(\tilde{p}_i, \tilde{p}_j) \le \delta$

$\Rightarrow$ $\tilde{y}_j \le \tilde{y}_i + \delta$

$\Rightarrow$ $\tilde{p}_j \in B$

$\overset{clm}{\Longrightarrow}$ $|B \cap S_\delta| \le 8$

$\overset{\tilde{p}_i, \tilde{p}_j \in B}{\Longrightarrow}$ $|i-j| \le 8$ in y-coord sorted order in S_δ ☒



thm: two-dimensional closest pair in $O(n \lg n)$

vs sketch idea: sorting on $\boxed{\text{every}}$ recursive call is wasteful

instead: sort P by x-coord $\boxed{\text{once}}$ [at beginning]
— " — y — " —

use to sort L, R in $O(n)$ time [vs $n \lg n$] [to support recursion]

correctness: same

complexity: $T(n) \le \underbrace{O(n \lg n)}_{\text{[init sort]}} + \underbrace{R(n)}_{\text{[recursion]}} \le O(n \lg n)$

$R(n) \le 2R(n/2) + O(n)$

$\le \cdots$

$\le O(n \lg n)$ ☒

rmk: two dimensional closest pair in $O(n)$ time, using $\boxed{\text{randomization}}$

today: divide and conquer: two dimensional closest pair

next lecture: dynamic programming

logistics: - pset 0 out F17
- signup -piazza -gradescope

view alg as reduction to ~1D
fix retired restriction
clarify counterexample