Michael A. Forbes
miforbes@illinois.edu
2024-03-07. 1

CS473 Algorithms : Lecture 15 (2024-03-07)   revise

Logistics : - pset 5 out
- pset 6 out F17

last lecture : - dictionary problem [insert, lookup] [array vs linked list] [no single hash function works]
hashing - fully random [good expected load] [costly to store]
- universal hashing [pseudo random - good expected load, cheap to store, evaluate]

today : closest pair   title
reading [?]   reading, lecture.

def. [lecture 2] given $p_1 = (x_1, y_1), \ldots, p_n = (x_n, y_n) \in \mathbb{Z}^2$, the closest pair problem is to find $\underset{\text{arg}}{\wedge} \min_{i \neq j} \text{dist}((x_i, y_i), (x_j, y_j))$ [value] value vs. arg.
$= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = d(p_i, p_j)$ [euclidean dist]

convention : unit cost arithmetic $\Rightarrow$ all integers are $O(\lg n)$ bits $\Rightarrow$ $|x_i|, |y_j| \le N = \text{poly}(n)$ [simplify notation]

wlog : $x_i, y_j \ge 0$ [clear] $\Rightarrow p_i \in \{0, \ldots, N\}^2$

thm : [lecture 2] closest pair in $O(n \lg n)$ deterministic time [nontrivial divide and conquer]

thm : [today] — " — in $O(n)$ expected randomized time [$n \lg n$ vs n] ["simple", more flexible]

idea : [data structure] [last lecture: dictionaries] : efficient way to store data

Q. can we verify whether $\min \text{dist} \ge \Delta$? [significant warmup] $\hookrightarrow$ - user data [hard choice]
- algorithmic data [intermediate results] [today]

idea : process points in order

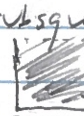
def. $\Delta_k = \min_{i,j \le k} d(p_i, p_j)$

Q. [exercise] Δ_n vs Δ? Δ_{k-1} vs Δ_k?

idea : coarse discretization of space

def. $\delta > 0, \delta \in \mathbb{R}$
a δ-subsquare is a subset of $\mathbb{Z}^2$ given by $S_{\alpha,\beta} = \{(x, y) \in \mathbb{Z}^2 : \alpha \le x < \alpha + \delta,\ \beta \le y < \beta + \delta\}$, $\alpha, \beta \in \mathbb{R}$ [open vs closed]

eg :



α, β

clm : $\Delta_k \ge \Delta \Rightarrow$ each $\Delta/2$ subsquare has ≤ 1 point from $p_1, \ldots, p_k$ [separated points] [≈ lecture 2]

pf. [two close points] $d((x,y),(z,w)) = \sqrt{\underbrace{(x-z)^2}_{\le (\Delta/2)^2} + \underbrace{(y-w)^2}_{\le (\Delta/2)^2}}$ vs strict.
$\le \sqrt{\Delta^2/2} = \Delta/\sqrt{2} < \Delta$ [close points] [but $\Delta_k \ge \Delta$] ⊠

[$\Delta_k \ge \Delta \Rightarrow$ rounding points to δ-subsquares loses no info]

idea : round coordinates [points to subsquares]

def. $(a, b) \in \mathbb{N}^2$. the (a,b) $\Delta/2$-subsquare of $\mathbb{N}^2$ [even vs Z] is $S_{a \cdot \frac{\Delta}{2}, b \cdot \frac{\Delta}{2}}$ [as above]
the $\Delta/2$-grid is $\{S_{a,b} : 0 \le a, b \le \frac{2N}{\Delta}\}$ [non-neg coord] $= T_{a,b}$

Michael A. Forbes
miforbes@illinois.edu
2024-03-07-2

def: [last lecture] a dictionary over $\mathcal{U}$ [universe] [either integers, strings, ...] is a data structure for storing a set $S \subseteq \mathcal{U}$ of keys x, along w/ associated values y

it supports: <u>insert</u>(x,y): add key x to S, w/ value y

<u>lookup</u>(x): decide if $x \in S$, if so return value y

· radar [complexity in terms of $n = |S|$]

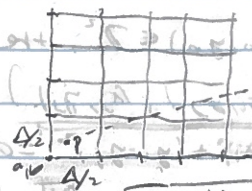
· dvsd

[converse to subsquare vs points]

prop: $d(p,q) \le \Delta$, $p \in T_{a,b}$, $q \in T_{c,d}$ [$\Delta/2$-grid]

$\Rightarrow |a-c|, |b-d| \le 2$

pf: by contrapositive [$\Delta/2$-grid]



[assume conclusion false]
[q wlog here]

$|a-c| > 2 \Rightarrow d(p,q) = \sqrt{(x-z)^2+(y-w)^2} \ge \sqrt{(x-z)^2}$

$= |x-z|$

$|a-c| > 2$

$> \Delta$

~~$|b-d| > 2 \Rightarrow$~~

[consider points in order, test for previous points]

algo:

- init dictionary A over $\mathcal{U} = \{(a,b) : 0 \le a,b \le \frac{2N}{\Delta}\}$ [$\Delta/2$-grid]
- for $i \le n$:

$(a,b) = \left(\left\lfloor \frac{x_i}{\Delta/2} \right\rfloor, \left\lfloor \frac{y_i}{\Delta/2} \right\rfloor\right)$ [round]

compute $\Delta_i' = \min_{\substack{p \in A[c,d] \\ |a-c| \le 2 \\ |b-d| \le 2}} d(p_i, p)$

· vis capacity

distmin

if $\Delta_i' < \Delta$, return p_i, p, Δ_i'

insert p_i into $A[a,b]$

return "$\Delta_n \ge \Delta$"

· vs 2022

prop: suppose $\Delta_i \ge \Delta$. then (a) algo reaches

(b) $A[a,b]$ empty at that moment [(a,b) not in dict]

pf: (a) algo doesn't reach only if find $(<\Delta)$-close pair within $p_1, \ldots, p_i \Rightarrow \Delta_i < \Delta$

(b) $A[a,b] = p_j$, $j < i \Rightarrow p_i, p_j$ in same $\Delta/2$-subsquare $\Rightarrow$ $(<\Delta)$-close $\Rightarrow \Delta_i < \Delta$

· correct

cor: suppose $\Delta_i \ge \Delta$ then for p_i - makes ≤ 25 dictionary lookups, [25 such (c,d)]

- returns ≤ 25 points [each cell has ≤ 1 point by above]

[complexity] [last lecture] [convention]

thm: $S \subseteq \mathcal{U}$, $|S| = n$, $|\mathcal{U}| = N \le \text{poly}(n)$

one can in deterministic $O(n)$ time construct a hash function family $\mathcal{H} = \mathcal{U} \to T$

- $|T| \le O(n)$
- choosing $h \in \mathcal{H}$ takes - $O(1)$ space to store
  - $O(1)$ time to eval
- insertion, lookup operations are $O(1)$ expected time

Michael A. Forbes, miforbes@illinois.edu
2024-03-07.3

binary search

cor: if $\Delta_n \ge \Delta$ [all dist $\ge \Delta$] then
- alg is correct
- runs in $O(n)$ expected time [+ hashing]

pf: uses dictionary on n subsquares in $N^2 \le \mathrm{poly}(n)$ size universe
[one per point, never reinsert, as $\Delta_n \ge \Delta$]
uses
- n insertions
- $\le 25n$ lookups [$\Delta_n \ge \Delta$, each lookup is single point]

$\Rightarrow O(n)$ dictionary ops $\Rightarrow O(n)$ expected time
linearity of expectation

cor: if $\Delta_{k-1} \ge \Delta$, $\Delta_k < \Delta$ [first dist $< \Delta$]
- alg is correct, returns $\Delta' = \Delta_k$ [no need to examine all points]
- runs in expected $O(k)$ time [as above]

pf: on $p_1, \ldots, p_{k-1}$: $\Delta_{k-1} \ge \Delta \Rightarrow$
- k insert
- $\le 25k$ lookup [all subsquares still hold ≤ 1 point]

p_k: 25 more lookups

$\Delta_k < \Delta \overset{\text{pigeonhole}}{\Rightarrow}$ will find p_i, $i<k$ - $d(p_i,p_k) = \Delta_k < \Delta$ [correct]

$\Rightarrow$ correct [finds violating points] $\Rightarrow$ will return it [also done]

$\Rightarrow O(k)$ time in expectation $\Rightarrow$ [no] insertion

thm: [so far] $\Delta > 0$ given. will either
- declare $\Delta_n \ge \Delta$ in $O(n)$ expected time
- find p_i, p_k with
  - $i<k$
  - $d(p_i,p_k) = \Delta_k < \Delta$

  in $O(k)$ expected time

Q: Δ unknown?

idea: try a Δ; [candidate]
- finish if $\Delta_n \ge \Delta$
- identify $\Delta' < \Delta$ [can only go down]

Q: how many times can Δ change?

A: can find examples w/ $\Theta(n)$ changes $\Rightarrow \Theta(n^2)$ runtime [worse than deterministic also]

[new idea needed]
idea: process points in $\boxed{\text{random}}$ order [randomness of dict] [randomness of order]

alg: randomly reorder $p_1, \ldots, p_n$

$(p,q) = (p_1,p_2)$ [candidate soln]

$\Delta = d(p,q)$

while

verify $\Delta_n \ge \Delta$ [existing alg]

if $\Delta_n \ge n$, return $(p,q), \Delta$

else $(p,q),\Delta \leftarrow (p',q'),\Delta'$ [new min dist, as found] [clearly]

prop: alg is correct

Q: complexity?

prop: alg can only update from $\Delta \to \Delta' = \Delta_k$ $\boxed{\text{once}}$ [Δ always goes down] [so does this happen?]

def: $I_k = \begin{cases} 1 & \text{verification alg updates from } \Delta \to \Delta' = \Delta_k \\ 0 & \text{else} \end{cases}$

prop: # dict ops of alg is $O(n + \sum_{k=1}^{n} k \cdot I_k)$

pf: $O(n)$ ops in final pass [must occur]

$O(k)$ ops to update $\Delta \to \Delta' = \Delta_k$, happens iff $I_k = 1$

[ranks only σ]

prop. $\Pr[I_k = 1] \leq 2/k$

pf : $I_k = 1$ iff update $\Delta \to \Delta_k$

⤷ was Δ_j some $j < k$

iff $\Delta_1, \dots, \Delta_{k-1} > \Delta_k$

iff $\min_{i,j \leq k} d(p_i, p_j) > \min_{i<k} d(p_i, p_k)$

$\Leftrightarrow$ random order placed either point of $\operatorname{argmin}_{i,j \leq k} d(p_i, p_j)$ in last position of first k points

eg : $p_1 \dots p_k$

$\Rightarrow \leq 2/k$ [two bad options for slot k in random order]

· vs 2022 midterm

· k rand var

cor : also runs in $O(n)$ expected runtime

[order] [dict] [dict ops cost] [O(k) dict ops] [k rand var] [k arg]

pf $\mathbb{E}_\sigma \mathbb{E}_A [D + \sum_k D_k \cdot I_k]$

⤷ dict ops ⤷ O(k) ops

$= \mathbb{E}_\sigma [\mathbb{E}_A[D] + \sum_k \mathbb{E}_A[D_k \cdot I_k]]$

⤷ dict ops, n [not] rand — $\underbrace{}_{O(n)\ [\text{linearity}]}$

⤷ not rand wrt A [only on σ]

⤷ $O(k)$ ops, k and I_k indep of A [expectation outside]

$= I_k \cdot O(k)$

$= \mathbb{E}_\sigma [O(n) + \sum_k O(k) \cdot I_k]$

[linear] $= O(n) + \sum_k O(k) \cdot \underbrace{\Pr[I_k = 1]}_{\leq 2/k}$ [indicator]

$\underbrace{}_{\leq O(1)}$

$\underbrace{}_{\leq O(n)}$

$= O(n)$

· vs 2022 midterm

today - closest pair

- $O(n \log n)$ deterministic [better divide and conquer]
- verification dist $\geq \Delta$ [linear points, check local subsquares via dict]
- randomly order points [low prob of many updates]

reading:

Lecture 1 : pages 5 a/c

pset 6 a/c F17