Closest pair

2017년 9월 25일 월요일
오후 3:55

problem : Closest pair of point n points in $\mathbb{R}^d$

⇒ let $S = \{p_1, p_2, \dots p_n\}$, find a pair p_i and p_j that the distance between p_i and p_j is mimimized (Euclidian dist : $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

- $O(n^2)$ Brute force.

- 1D
  1. Sort $O(n \log n)$
  2. Scan in order : $O(n)$

  □ $O(n \log n)$

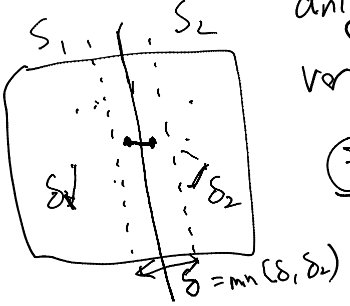
- 2D

  if $|S| = 1$ then return $(p_1, \text{null}, \infty)$
  if $|S| = 2$ then return $(p_1, p_2, \text{dist}(p_1, p_2))$
  else

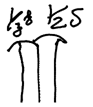
  $O(n)$ ① partition S into S_1' and S_2' about the median x-value x_m ~~recursively~~

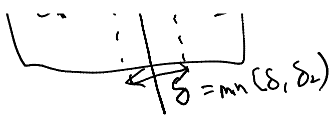
  ② recursively call procedure on S_1 S_2 yielding
  $(p_{11}, p_{12}, \delta_1)$: closest pair in S_1 with distance δ_1
  $(p_{21}, p_{22}, \delta_2)$: 〃 S_2 〃 δ_2
  $\delta = \min(\delta_1, \delta_2)$

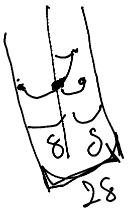
  Create subset S_3' st it contains only points within 2δ-wide vertical stripted center

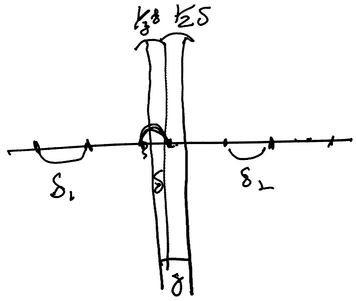


③









③ from S_3 find closest pair s.t
$d(p_i, p_j)$ is minimal $p_i \in S_1$ & $p_j \in S_2$

endif
if closet pair is in S_1 or S_2 then return
2017-09-25 오후 4:20에 이슬이(가) 수정
output accordingly
elseif return closet pair in the Boundary

* finding closest pair facing the boundary

$O(n\log n)$: Assume the points are sorted by y-values.

- Let S_3 be the set of points in S whose x values are within δ of x_m
- Let the y-value of the points in S_3 be sorted $(y_1 \le y_2 \le y_3 \dots \le y_n)$
- Defind interval that covers the y-values as follows :

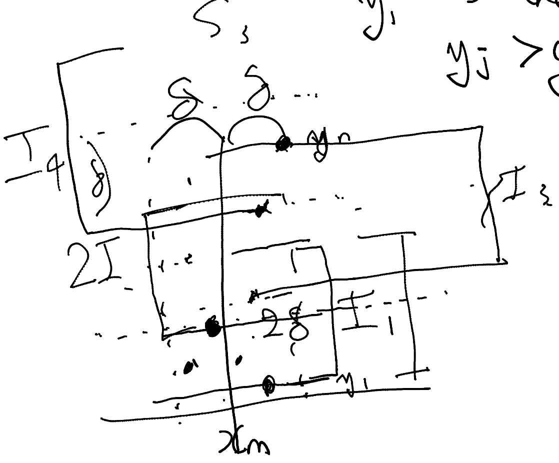
$$I_1 = [y_1, y_1 + 2\delta) \text{ : initial interval}$$

Given $I_k = [y_i, y_i + 2\delta)$ for some $k \ge 1$
if $(y_i + \delta \le y_n)$ the define — termination condition

$$I_{k+1} = [y_j, y_j + 2\delta)$$

S_3 y_j is the smallest y-value s.t.
$y_j > y_i + \delta$

y_j is the smallest y-value s.t. $y_j > y_i + \delta$



* obsersveration

① each point fall in one or two intervals

② If point p' and p'' have
→ $|y' - y''| < \delta$ then there is an interval I that s.t. $p', p'' \in I$

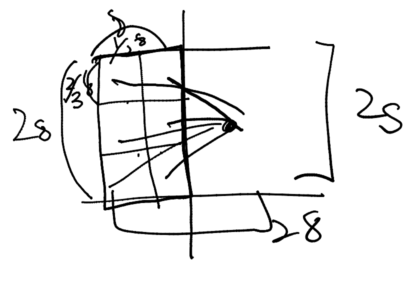
③ there are not too many points in a interval.

For each interval I, compare each point where y-value $\in I$ with each other points

⇒ $O(n)$

* verify max # of points in a interval

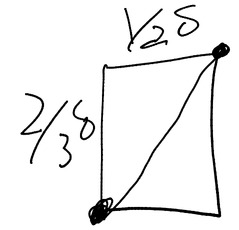
⇒ Divide $\delta \times 2\delta$ rectangle K in 6 of $\delta/2 \times 2/3\,\delta$ subrectangle



for each point outside $\delta \times 2\delta$, at most 6 comparisons are required

$1/2\,\delta$

at most 6 comparisons are required



$$\sqrt{\left(\tfrac12\delta\right)^2+\left(\tfrac23\delta\right)^2} = \delta\sqrt{\tfrac{25}{36}} \le \underset{\sim}{\tfrac56}\delta < \delta$$

⇒ at most 6 points per
$\delta \times 2\delta$ rectangle.

⇒ time
$O(1)$ for each interval

$O(n)$ time for all intervals.

Time for closest pair Alg in 2D

① initial sort $O(n\lg n)$

② recurrence step $\begin{cases} T(n) \le C & n \le 2 \\ T(n) \le \underset{\sim}{Cn} + T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) & n > 2 \end{cases}$

$O(n\lg n)$

$\therefore O(n\lg n)$