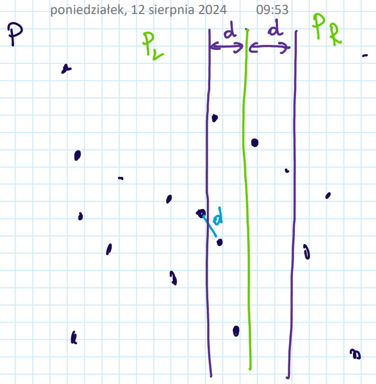
Para najbliżej położonych punktów

poniedziałek, 12 sierpnia 2024 09:53



$n = 2^k$

$p_i(x_i, y_i)$

$P = p_1, \dots, p_n$

Szukamy p_i, p_j - najbliżej położone

Z czym walczymy?

par punktów = n^2 ← rozwiązanie naiwne

Chcemy $T(n) = T(n_1) + T(n_2) + \Theta(n)$
podzielić na dwa podzbiory

Dzielimy punkty prostą na dwa zbiory
Jeżeli któryś z punktów leży na prostej
to dodaj go do dowolnego zbioru

Dokonamy podziału: sortujemy punkty
po współrzędnej x-owej, połowę dodajemy do
jednego zbioru, a drugą do drugiego

Rekurencyjnie wywołujemy się na lewej i prawej stronie
i szukamy pary najbliżej położonych punktów

Teraz musimy rozważyć wszystkie pary ($\frac{n^2}{4}$), które
mają swój jeden punkt w pierwszym zbiorze, a drugi w drugim

Wtedy $T(n) = 2T(\frac{n}{2}) + \frac{n^2}{2}$ ← rekurencja bierze w łeb

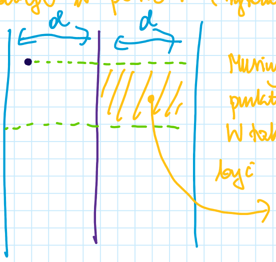
Ale zauważmy, że jest sens rozważać tylko punkty odległe o
d od pasa (bo d to obecnie najmniejsza odległość jaką udało nam
się osiągnąć)

Ile jest takich par? KONTRPRZYKŁAD

bo w pasie o szerokości d wciąż leżą wszystkie punkty

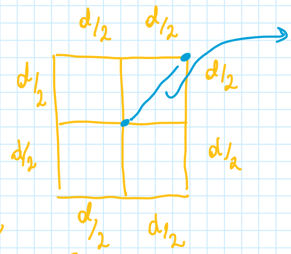
Ale możemy trochę inaczej analizować punkty w pasie

Co się dzieje w pasie? (wyrażone na zbliżeniu)



Musimy rozważać tylko punkty odległe o d
W takim kwadracie mogą leżeć nie więcej niż 4 punkty

Czemu? One wszystkie leżą po jednej stronie środkowej linii, gdyby było ich więcej to odległości między nimi byłyby mniejsze niż d



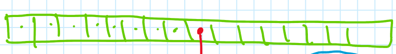
$\frac{\sqrt{d^2}}{2} \leq d$

zatem dwa punkty nie mogą leżeć w tym samym podkwadracie

zoom na jeden kwadracik

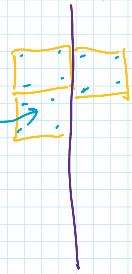
Idziemy tym pasem o rozmiarze 2d od góry

Więc teraz posortujemy te punkty po współrzędnej y-owej



tablica punktów z tego pasa posortowana według y

ten punkt jest już na pewno z kwadracika, którego nie będziemy rozważać



Aby p posortowane względem y kolejno

Zatem punktów które chcemy sprawdzić jest max 7

W ten sposób otrzymamy procedurę

- sortowanie po x_{sorg} by dokonać podziału na dwa zbiory

n log n

$$T(n) = n \cdot \log n + 2 \cdot T\left(\frac{n}{2}\right) + n \log n = 2T\left(\frac{n}{2}\right) + \Theta\, n \log n = \Theta(n \log^2 n)$$

wywołanie na dwóch zbiorach rozłącznych

sortowanie po y by wydobyć te 7 interesujących nas punktów

[illegible]

Ale możemy te punkty posortować raz przed działaniem algorytmu (po np x_{sorg})

A następnie po y_{sorg} też przed działaniem algorytmu

Dzięki temu będziemy mogli wyłuskiwać elementy z tablicy bo będą już w dobrej kolejności

Zatem ostateczne złożoności: $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$ czyli $O(n \log n)$