

Divide and Conquer III

Michael Tsai

2012/9/27

Closest Pair of Points

- 問題: 在 $n \geq 2$ 個點中(集合 Q), 找出一對點其兩點之間的距離為最小. 每個點為一二維座標.
- 距離: 以Euclidean distance定義.
- $p_1 = (x_1, y_1)$ $p_2 = (x_2, y_2)$
- $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- Q 中可能有一樣的兩點 (則其距離為0)
- Reference: (Textbook 33.4, p.1039)

Closest Pair of Points - 暴力法

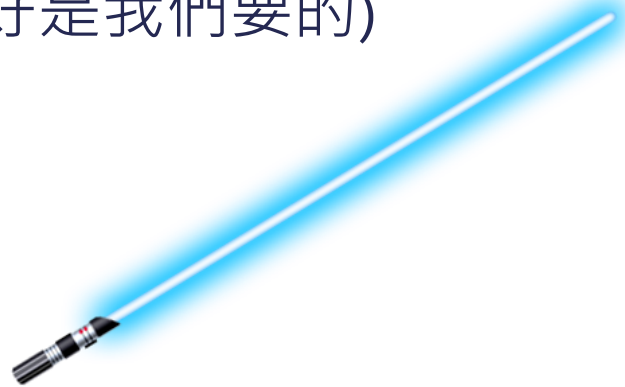
- 算出每一對的距離
- $\binom{n}{2} = \Theta(n^2)$

Closest Pair of Points – D&C

- 思考流程:
- 原本是 $\Theta(n^2)$, $\Theta(n \log n)$ 似乎是個不錯的目標
- 大刀先拿來砍一砍.
- 大概會長這樣: $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(??)$
- 可以對Q中的點排序嗎?
 - 選擇一: 開始的時候先全部排一次
 - 花 $\Theta(n \log n)$, 跟目標一樣...ok的!
 - 選擇二: 每一次遞迴呼叫都對點排序
 - 那遞迴式至少變成: $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n \log n)$
 - $T(n) = \Theta(n \log^2 n)$
(使用課本4.6-2 變形master theorem) not ok!

Closest Pair of Points - D&C

- $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$ 的話...
- $T(n) = \Theta(n \log n)$ (正好是我們要的)
- Algorithm組成元素:
 - 一開始可以對點先排序
 - 光劍一隻:
 - Divide的時候切成兩份等份
 - Combine的時候最多可以花 $\Theta(n)$ 的時間

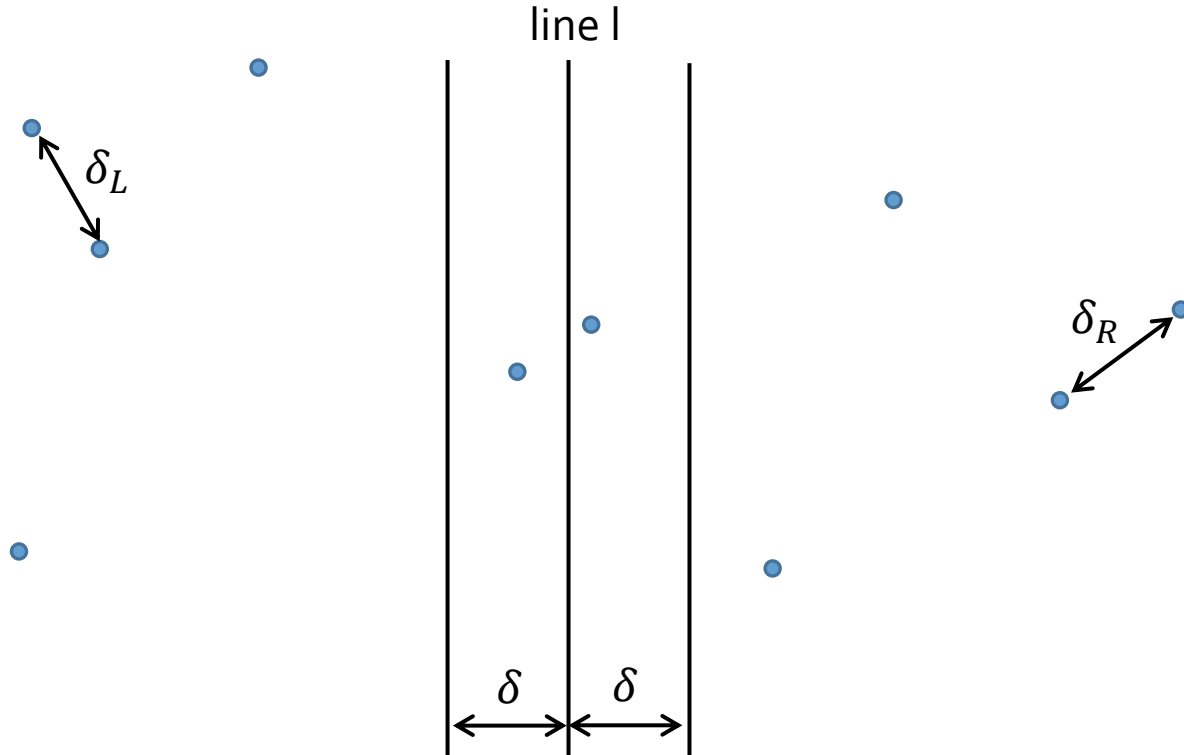


Closest Pair of Points – D&C

- 先想想Recursive case
- Divide:
 - 劃一條直線 l , 把所有的點分成兩部分 P_L 及 P_R , 使 $|P_L| = \lfloor \frac{|P|}{2} \rfloor$ and $|P_R| = \lfloor \frac{|P|}{2} \rfloor$, P_L 中的點都在 l 上或它的左邊, P_R 的點都在 l 上或它的右邊
- 遞迴呼叫會解決找出 P_L 及 P_R 中的closest pairs, 假設找到的最短距離分別為 δ_L, δ_R , and $\delta = \min(\delta_L, \delta_R)$
- Combine:
 - Closest Pair有三種可能:
 - P_L 或 P_R 中找到的
 - 或者是一個點在 P_L 中, 一個點在 P_R 中的, 而兩點距離小於 δ

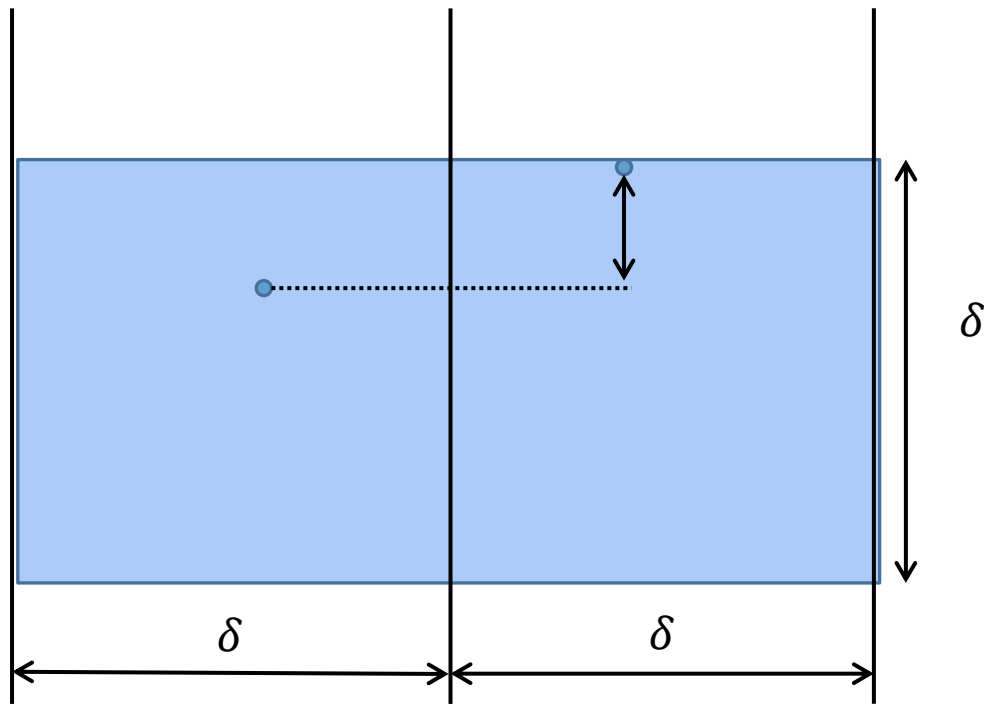
Closest Pair of Points – D&C

- 怎麼找出一點在 P_L , 一點在 P_R , 而兩點距離 $< \delta$?
- 只考慮可能的點: 在 2δ 範圍內的點



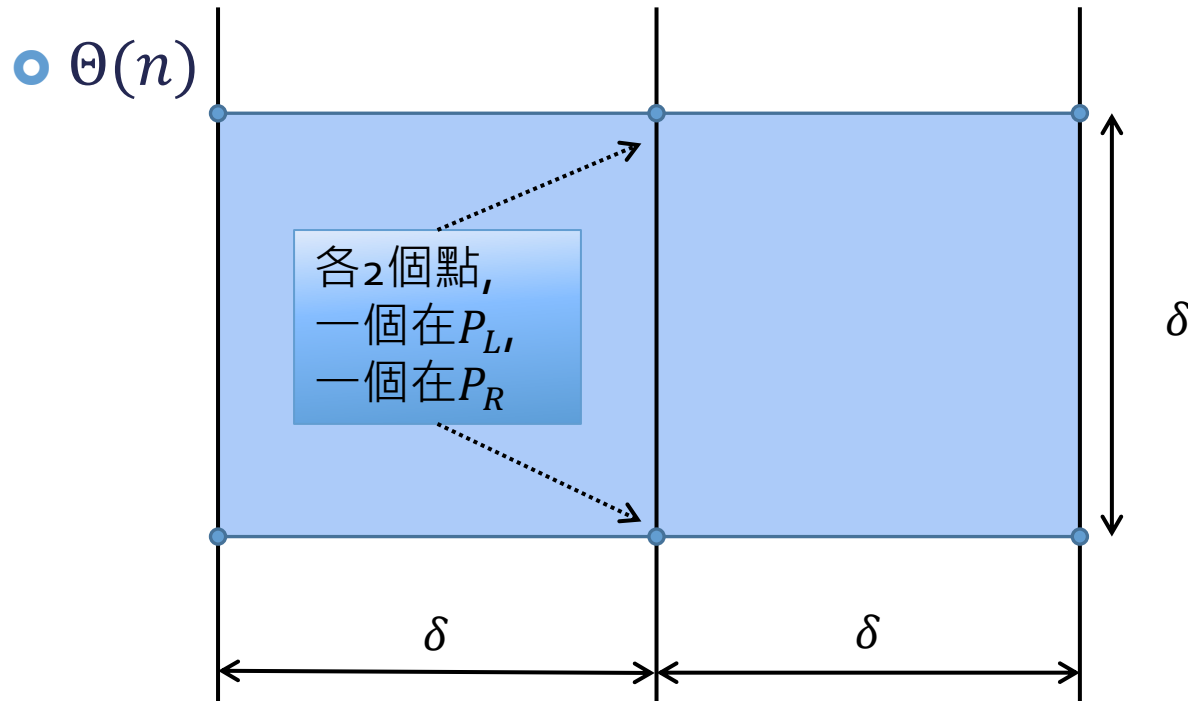
Closest Pair of Points – D&C

- 可是combine只能花 $\Theta(n)$ 喔 (不能暴力找每個pair)
- 關鍵: 2δ 長條範圍內不需要找每個pair!
 1. 構成closest pair的兩點一定在 $2\delta \times \delta$ 的長方形內



Closest Pair of Points – D&C

2. $2\delta \times \delta$ 的長方形內最多可以放下8個點 (點和點之間的距離不可以小於 δ , 否則就矛盾了)
- 因此對每個在 2δ 長條中的點, 如果照Y排序過後, 只需要檢查它和它後面七個點的距離!



Closest Pair of Points – D&C

- Base case:
- 當 $n \leq 3$ 時, 則直接用暴力法找出closest pair
- 最多算三次距離.
- $\Theta(1)$

Closest Pair of Points – D&C

○ Divide:

 $\Theta(n)$

- 劃一條直線 l , 把所有的點分成兩部分 P_L 及 P_R , 使 $|P_L| = \lfloor \frac{|P|}{2} \rfloor$ and $|P_R| = \lfloor \frac{|P|}{2} \rfloor$, P_L 中的點都在 l 的左邊, P_R 的點都在 l 上或它的右邊

- 遞迴呼叫會解決找出 P_L 及 P_R 中的closest pairs, 假設找到的最短距離分別為 δ_L, δ_R , and $\delta = \min(\delta_L, \delta_R)$

 $2T\left(\frac{n}{2}\right)$

○ Combine:

- 將在 P_L 及 P_R 中不在直線 l 的左右 δ 距離內的點去除, 得到 P_L' 及 P_R' .

- 對於 P_L' 及 P_R' 中的點, 檢查它和Y座標在它後面的七個點的距離是否小於 δ . $\Theta(n)$

- 若是的話則此為closest pair. 若否的話則遞迴呼叫所得到的 δ 距離的pair為closest pair.

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$



Closest Pair of Points – D&C

○ Divide:

- 劃一條直線 l , 把所有的點分成兩部分 P_L 及 P_R , 使 $|P_L| = \lfloor \frac{|P|}{2} \rfloor$
and $|P_R| = \lceil \frac{|P|}{2} \rceil$, P_L 中的點都在 l 的左邊, P_R 的點都在 l 上或它的右邊

- 遞迴呼叫會解決找出 P_L 及 P_R 的最短距離分別為 δ_L, δ_R , and $\delta = \min(\delta_L, \delta_R)$

需要所有P中的點照x座標排序

○ Combine:

- 將在 P_L 及 P_R 中不在直線 l 的左邊及右邊中的點分別放入 P_L' 及 P_R' .
- 對於 P_L' 及 P_R' 中的點, 檢查它和Y座標在它後面的七個點的距離是否小於 δ .
- 若是的話則此為closest pair, 若不是的話則遞迴呼叫所得到的 δ 距離的pair為

需要所有P中的點照y座標排序

但是不能每個遞迴呼叫都排序!
如何用低於 $\Theta(n \log n)$ 的時間取得排好序的點?

Closest Pair of Points – D&C

未排序的P

依照y座標排序的P

依照x座標排序的P

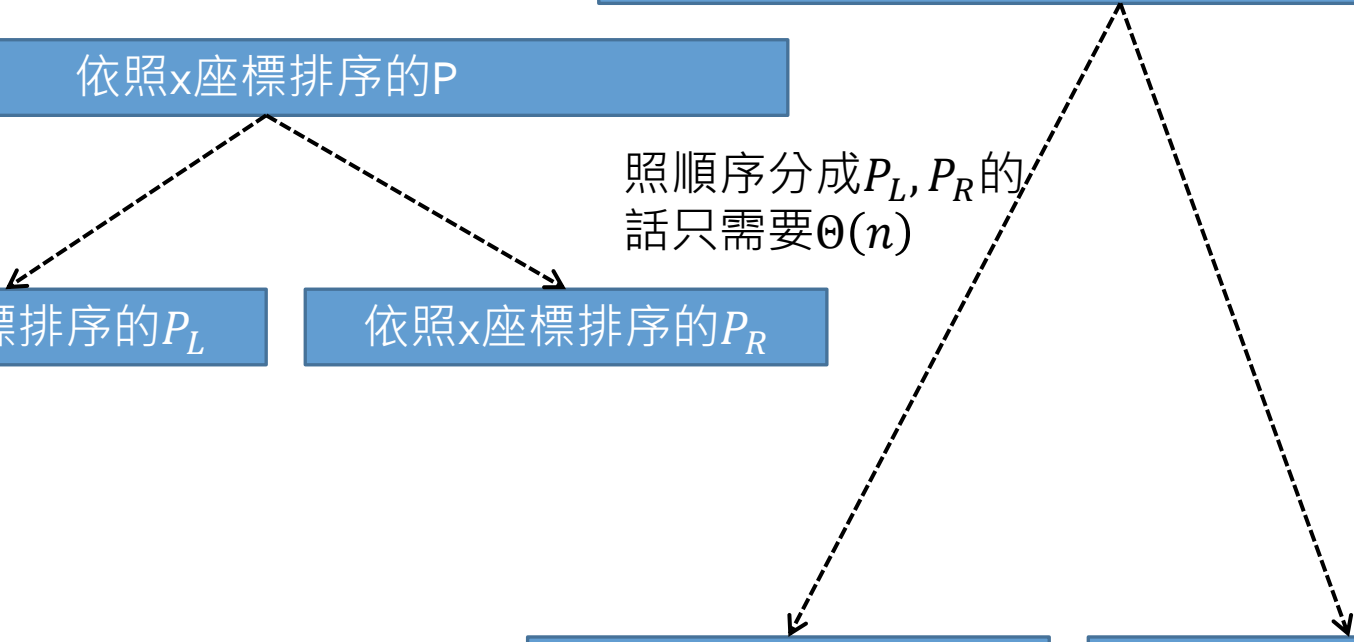
照順序分成 P_L, P_R 的
話只需要 $\Theta(n)$

依照x座標排序的 P_L

依照x座標排序的 P_R

依照y座標排序的 P_L

依照y座標排序的 P_R



Closest Pair of Points – D&C

- 不包含一開始的sorting的話:
- $T(n) = \Theta(n \log n)$
- 包含所有的話:
- $T'(n) = \Theta(n \log n) + T(n) = \Theta(n \log n)$

Sorting所花時間



休息一下 - 程式設計師常見藉口

- by Santos Liu on Saturday, March 5, 2011 at 1:00am
- 第 20 名：這很奇怪喔。
- 第 19 名：以前從來不會這樣啊！
- 第 18 名：昨天明明會動的啊！
- 第 17 名：怎麼可能~
- 第 16 名：這一定是機器的問題。
- 第 15 名：你到底是打了什麼才讓程式當掉的？
- 第 14 名：一定是你的資料有問題。
- 第 13 名：我已經好幾個禮拜沒碰那一段程式了。
- 第 12 名：你一定是用到舊版了。
- 第 11 名：一定是巧合！為什麼這種壞運氣只讓你碰上。
- 第 10 名：我不可能什麼功能都測試到吧，有 bug 是正常的！
- 第 9 名：這個不可能是那個的原始碼！
- 第 8 名：這程式應該是會動的，只是我寫好後還沒做測試。
- 第 7 名：可惡！一定有人改了我的程式。
- 第 6 名：你有檢查過你的電腦有沒有病毒嗎？
- 第 5 名：儘管這功能還不能動啦，你覺得他如何？
- 第 4 名：在你的系統不能用那一個版本的程式啦！

- 第 3 名：你幹嘛要那樣操作，都是你的問題。
- 第 2 名：程式發生問題時你在哪裡？
- 第 1 名：在我的機器明明就可以動啊！