

專題1

Maximal Points

資工二A 110502518 陳文獻

題目敘述

- 在一個 N 維空間中有 n 個點，任取兩個點 p, q ，當 $p.x \geq q.x$ 、 $p.y \geq q.y$ 、 $p.z \geq q.z$ 皆成立時稱為 p dominates q 。
- 當一個在點集合 S 中的 p 點不被任何其他屬於 S 的點 dominates 時該 p 點就被稱作為一個 maximal point，目標是要將所有屬於 S 的 maximal points 找出來。

二維的做法(X, Y)

- 先依照 X 座標, Y 座標的順序將點降序排序。
- 遍歷串列
- 持續記錄 Y 座標的局域最大值，未使最大值更新的點皆是可被`dominate`的點。

三維的做法(X, Y, Z)

- 按照X座標、Y座標、Z座標降序排序串列 (這麼做可以確保遍歷串列時取到的點都不會被之後的點dominate)
- 建立一個樹狀數組(BIT)，以離散化的Y座標值作為索引，區間內最大Z值作為值。
- 以此結構存入所有不被dominate的點，每次得到新的點 P_i 時，查詢BIT中， P_i 的Z值是否大於Y座標小於 P_i 的Y座標的條件下，最大的Z值。(即 P_i 是否不被拜訪過的點dominate)
- 複雜度分析: (n為輸入點個數)
- 排序 $O(n \log n)$ +遍歷點 $O(n)$ * 樹狀樹組操作 $O(\log n) \Rightarrow O(n \log n)$

pseudocode

Ans:= empty list

sorted_list := points sort by (X,Y, Z, NonIncreasing)

BIT:= list with initial value $-\text{INF}$

discretize Value of Y

For (i : 0 to N):

 quire BIT to get max_z (for all points which Y value is smaller then sorted_list[i].y)

 if (sorted_list[i].z > max_z):

 Ans.push(i)

 modify BIT with sorted_list[i]

Then we can get index of answer in sorted_list

複雜度實測

在測資長度 10^4

每個座標值(x, y, z)在0-100隨機的情形

