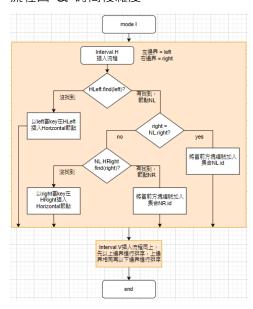
演算法導論 Lab1_report_B101016 曹寓恆

1. 資料結構:

見右圖·Horizontal 負責儲存 x 軸上區間的關係· Vertical 負責儲存 y 軸上區間的關係·再由 Interval 合成 完整的 rectangle 信息。

2. 流程圖 & 時間複雜度



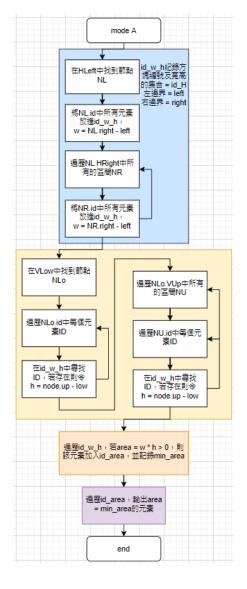
Mode I:

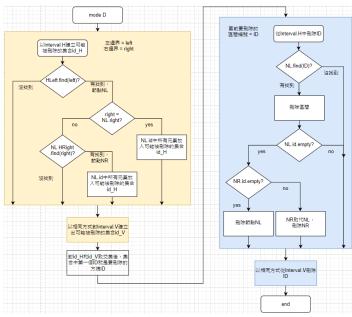
見上圖·Interval.Horizontal 中·在 HLeft 尋找插入點和 在 HRight 尋找插入點都需要 O(lgn)·處理 Interval.Vertical 也是·整體時間複雜度為 O(lgn)。

Mode A:

見右圖·藍色部分建立 id_wh 的時間複雜度為 $O(lgn+c+klgk)\cdot c$ 表示 NL 及 NR 的 id 總數·k 表示 id_wh 的大小·區間夠分散時時間複雜度約等於 $O(lgn+klgk)\cdot d$ 表示 NLo 及 NU 的 id 總數·橘色和紫色部分時間複雜度為 $O(k)\cdot$ 通常 k<<n 且 $c\approx d<<n$ · 區間夠分散時整體時間複雜度為 $O(lgn+klgk)\cdot d$ 表示 Id 的 Id Id 的 Id

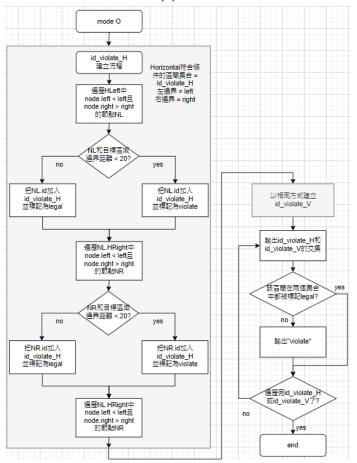
```
struct Horizontal // interval tre
{
    int left, right;
    set<int> id;
    map<int, Horizontal> HRight;
};
struct Vertical // interval tree
{
    int low, up;
    set<int> id;
    map<int, Vertical> VUp; // st
};
struct Interval
{
    map<int, Horizontal> HLeft;
    map<int, Vertical> VLow;
};
```





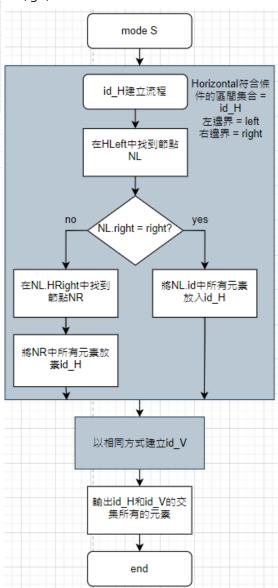
Mode O:

見下圖灰色部分‧找到所有 overlap 候選人的時間複雜度 為 O(n)‧輸出的部分的時間複雜度為 O(k)‧通常 k << n n 整體的時間複雜度為 O(n)。



Mode D:

見左圖黃色部分找到被刪除的候 選人的時間複雜度為 O(lgn)·藍 色部分刪掉區間的時間複雜度為 O(1)·整體的時間複雜度為 O(lgn)。



Mode S:

見上圖深灰色部分·找到所有候選人並建立集合 id_H 的時間複雜度為 O(lgn)·id_V 也是·輸出 id_H 和 id_V 交集元素的時間複雜度為 O(klgk)·k 為候選人集合的大小·通常 k << n·整體時間複雜度為 O(lgn + klgk)。