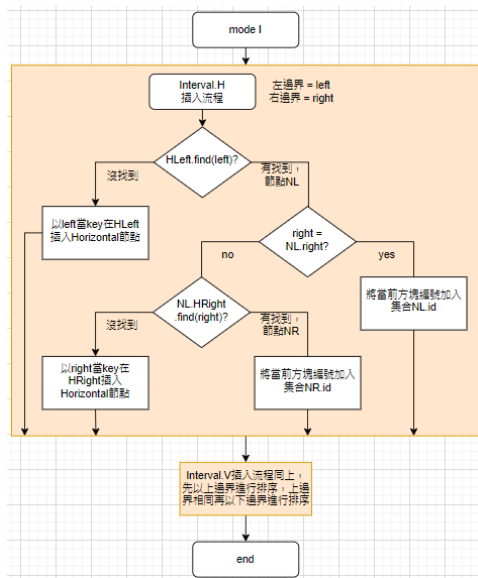


演算法導論 Lab1_report_B101016 曹寓恆

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 $O(\lg n)$ ，處理 Interval.Vertical 也是，整體時間複雜度為 $O(\lg n)$ 。

Mode A:

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

```

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

