

物件導向程式設計及應用第二次作業

Due: 2016/11/7 13:00

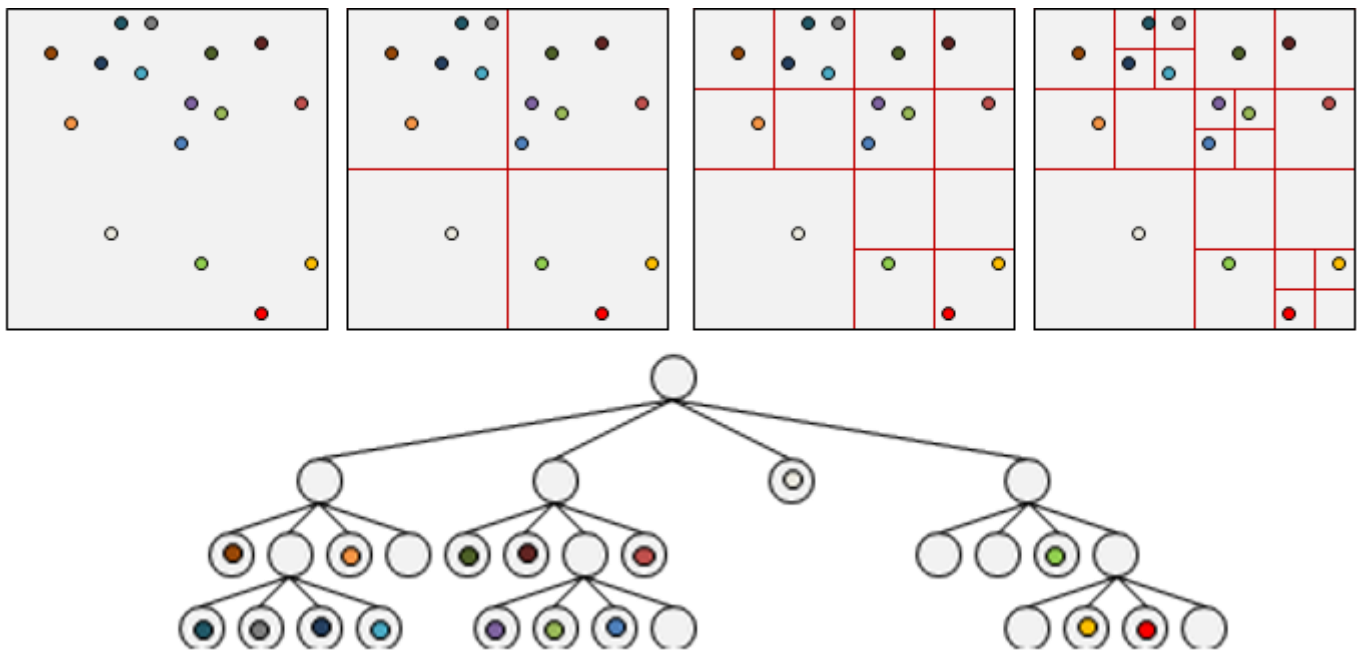
※注意事項：請依照課程網站內所公告之“作業檔案命名規則與規定”進行作業檔案命名以及繳交作業，未依照規定將斟酌扣分。

樹[1]，是一種抽象數據類型或是實作這種抽象數據類型的資料結構，用來模擬具有樹狀結構性質的數據集合。定義為由一個或數個節點所組成的有限集合，並包含幾個特性：

- 1.每個節點有零個或多個子節點；
- 2.沒有父節點的節點稱為根節點；
- 3.每一個非根節點只有一個父節點；

樹的種類可分為無序樹與有序樹，而有序樹中最常見的為二元樹。二元樹顧名思義任何一個節點最多只會有兩個分支，亦可以沒有分支或只有一個分支。

本次作業要試建立四元樹之資料結構，將附件(Point_HW2.txt)之點資料讀入程式中，並建立四元搜尋樹，並輸入點座標，透過四元樹搜尋離該點最近的點。如圖一所示，將平面分割為四等分，若區域內有一個以上的點，則繼續將該區域分為四等分，直到所有區域只有一個點為止。



圖一 四元樹示意圖(圖片來源：參考資料[2])

本次作業為了方便運算，平面上直角坐標系的邊界（即圖一之黑色框）， x 與 y 最大與最小值為 ± 100 ，附件之點資料亦不會超出此範圍。而根節點之分割中心固定為 $(0, 0)$ ，將根節點分割為四等分，其四個子節點的分割中心為 $(50, 50)$ 、 $(-50, 50)$ 、 $(50, -50)$ 、 $(-50, -50)$ ，以下節點亦同。另外，附件之最上方會新增點數目的資訊，讀檔時請注意。

[1]Wikipedia; "Tree (data structure)," on-line at [https://en.wikipedia.org/wiki/Tree_\(data_structure\)](https://en.wikipedia.org/wiki/Tree_(data_structure)), accessed on 6 Oct 2016.

[2]國立臺灣師範大學資訊工程學系;"演算法筆記- Region," on-line at <http://www.csie.ntnu.edu.tw/~u91029/Region.html>, accessed on 17 Oct 2016.

類別宣告如下：

```
class Point
{
private: // #1
    float x;
    float y;

public:
    Point(); // #2
    Point(float, float); // #3
    ~Point(); // #4

    void Set_data(float, float); // #5
    float operator[](int i) const; // #6
    friend std::istream& operator>>(std::istream&, Point&); // #7
    friend std::ostream& operator<<(std::ostream&, const Point&); // #8
};

class QuadtreeNode
{
private:
    const Point* data; // #9
    QuadtreeNode* nextNode[4]; // #10
    const Point separate_point; // #11
    const float size; // #12

public:
    QuadtreeNode(); // #13
    QuadtreeNode(const Point&, const Point&, const float); // #14
    QuadtreeNode(const QuadtreeNode&); // #15
    ~QuadtreeNode(); // #16

    bool InsertPoint(const Point&); // #17
    ..... FindClosestPoint(.....) const; // #18
};
```

[1] Wikipedia; "Tree (data structure)," on-line at [https://en.wikipedia.org/wiki/Tree_\(data_structure\)](https://en.wikipedia.org/wiki/Tree_(data_structure)), accessed on 6 Oct 2016.

[2] 國立臺灣師範大學資訊工程學系; "演算法筆記- Region," on-line at <http://www.csie.ntnu.edu.tw/~u91029/Region.html>, accessed on 17 Oct 2016.

※說明：

- #1.類別 Point，有兩個私有成員 x 與 y 表示二維空間中的點座標
- #2.基底建構子，將點預設為(0, 0)
- #3.兩個參數建構子
- #4.解構子
- #5.設定資料值，如有需要可使用
- #6.Subscript operator，[0]為 x、[1]為 y
- #7.重載輸入導向運算子，請於讀檔時使用
- #8.重載輸出導向運算子
- #9.記錄此節點的點資料，若無則指向 nullptr
- #10.指向此節點的四個子節點，若無則指向 nullptr
- #11.記錄此節點的分割中心
- #12.記錄此節點的分割邊界長度
- #13.基底建構子，於本類別將不使用，請調整使其無法被呼叫
- #14.三個參數建構子
- #15.複製建構子
- #16.解構子，釋放動態空間
- #17.將點傳入並插入四元樹中
- #18.將點傳入，尋找傳入點所屬的子節點，並將已屬於該子節點的點回傳
(引數與回傳值請自行設計)

請於主程式建立四元樹的根節點（命名為 Quadtree），並將讀入的點陣列依序利用 Quadtree 之成員函式 InsertPoint 插入四元樹中，以建立四元樹。之後，提示使用者輸入一點座標（請檢查輸入型態是否符合要求，若否則顯示錯誤訊息並結束程式），並利用成員函式 FindClosestPoint 找出輸入點所屬子節點內的點座標並使用輸出導向運算子列印其至小數點下 2 位。最後，請使用複製建構子複製四元樹，並同樣使用成員函式 FindClosestPoint 找出輸入點所屬子節點內的點座標並使用輸出導向運算子列印其至小數點下 2 位，兩者列印出的座標應相同。

另外，請於 console 視窗回答意下問題：以上兩類別中，有哪些成員函式可以直接使用編譯器預設的版本？為什麼？

- 請將上述宣告撰寫於 HW02.h 標頭檔案內。
- 請將上述宣告的定義撰寫於 HW02.cpp 程式碼檔案內。
- 請撰寫一個主程式 useHW02.cpp，以測試上述功能。

※不可在類別內加入其他成員或成員函式以及增加新的類別※

※除輸入與輸出導向運算子外，不可宣告任何函式與類別為夥伴(friend)※

※請勿使用標準樣板函式庫(Standard Template Library)或巨集指令※

[1]Wikipedia; "Tree (data structure)," on-line at [https://en.wikipedia.org/wiki/Tree_\(data_structure\)](https://en.wikipedia.org/wiki/Tree_(data_structure)), accessed on 6 Oct 2016.

[2]國立臺灣師範大學資訊工程學系; "演算法筆記- Region," on-line at <http://www.csie.ntnu.edu.tw/~u91029/Region.html>, accessed on 17 Oct 2016.