**HW2** 0416206李明峻

1. 環境 : 工作站(bsd1.cs.nctu.edu.tw)，使用C++撰寫
2. 程式說明 : 根據RB Tree的特性先宣告兩個class，分別是node以及RBT，而node就是樹的每個節點，而RBT則是存放RB Tree相關調整處理的函式，以及根節點(root)和樹葉節點(neel)。總共有兩種方式:Insert和Delete，InsertRBT(int key)這個函式先對新增的node做初步的判斷以及調整，run至底部後必須將RB Tree做更進階的調整，方法在InsertRBTfixedup(node \*current)這個函式內，根據不同的case做不同的調整 : case1 : 新增節點的uncle node若是紅色的，進行調整；case2 : 新增節點的uncle node是黑色的且新增節點是parent的rightchild；case3 : 新增節點的uncle node是黑色的且新增節點是parent的leftchild；case2可能演變成case3，結束case3之後即可滿足RBT之特性。而對於Delete，DeleteRBT(int key)這個函式一樣先進行deletedNode的檢查及調整，而DeleteRBTfixedup(node \*current)則進行delete的細部調整，一樣根據不同的case進行調整，case1 : deletedNode的sibling若為紅色，調整顏色並leftRotate；case2 : deletedNode的sibling為黑色且sibling的兩個child皆為黑色，進行調整；case3 : deletedNode的sibling為黑色且sibling的rightchild為黑色而leftchild為紅色；case4 : deletedNode的sibling為黑色且sibling的rightchild為紅色。RBT可能經過case2的修正就滿足RBT特性，也可能經過case3連接case4後滿足RBT特性。

經過這些樹的調整及修改，最後再將樹由中序traverse印出來即完成。