CS Data Structure

Fall 2023

Program Exercise #6

In-class Demo: 2023/11/30 (Thu.) 13:10-16:00

* Submit Due Date: 2023/12/4 (Mon.) 23:59:00

- 當天交滿分 100，每隔一天扣 10

Problem: 用 binary search tree & min heap 來模擬拍賣網站的交易行為

模擬一個拍賣網站的交易行為，當顧客欲購買某項商品時，希望能夠快速搜尋到所需的商品項目，並且得知目前該商品最低標價為多少。本次作業希望同學使用Binary search tree及 heap 的架構來實作一個符合上述要求的程式。

每項商品有以下資料：
1. 商品名稱：(英文，不論大小寫只要字元皆相同即視為同一個商品)。
2. 此商品下的所有賣家資料。

賣家資料：
1. 賣家ID：(英數字皆可)。
2. 商品價格。

為了簡化拍賣的情況，假設每項商品最多只有 7 個賣家。

Basic requirements

1. **insert**: 新增商品及賣家資料，先執行search binary search tree的動作，若
   (1) 有此商品名稱，只需將賣家資料新增至商品項目下，賣家資料依據價格高低使用min heap 的方式存放。
   (2) 無此商品資料，則將商品以A-Z的順序(大小寫視為相同)新增到binary search tree 以及將賣家資料新增到min heap 中。
   Format:

指令	商品名稱	賣家 ID	商品價格
insert	CD	John	280

2. **search**: 搜尋商品名稱，若
   (1) 搜尋成功則將此商品下所有賣家資料(賣家ID、商品價格)存在searchTable中。
   (2) 搜尋失敗則將錯誤訊息存在searchTable中。
   Format:

指令	商品名稱
search	CD

3. **buy**: 購買商品名稱，首先必須search binary search tree，若
   (1) 有此項商品，則假設買家皆以商品價格為考量，並且選擇商品價格最低廉者購買，即取出min heap的root，然後將成交的賣家資料(賣家ID、商品價格)存在buyTable中，並且刪除此賣家的資料，並將賣價最低的賣家移至root，假設已經沒有賣家在min heap中，則從binary search tree刪除此商品。
   (2) 無此項商品則將錯誤訊息存在buyTable中。
   Format:

指令	商品名稱
buy	CD

4. **sort**: 使用inorder traversal 將商品名稱依照(A-Z)順序一一將結果存在sortTable中。
   Format:

指令
sort

5. **report**: report 後必須產生4 個files (searchTable.txt, buyTable.txt, sortTable.txt, LogTable.txt)，詳見下頁表格。
   Format:

指令
report

以上都必須使用Binary Search Tree的insert, search, delete, traversal 的演算法及min heap 的insert, delete來實作，否則將沒有分數(例如:search 使用暴力法搜尋)，各個Tables 說明詳見下文。

Input File Format (實際的 input file 不會有 comment //….):

```
Insert CD Mary 280          //新增 CD 這個商品以及賣家 Mary 賣此商品 280 元
insert TV John 20000        //新增 TV 這項商品以及賣家 John 賣此商品 20000元
buy Book                    //購買 Book 這項商品
insert Book Jacky08 220     //新增 Book 這項商品及賣家 Jacky08 賣此商品 220元
search CD                   //搜尋 CD 的所有賣家資料
insert Book Jacky01 200     //新增 Book 這項商品以賣家 Jacky01 賣此商品 200元
insert Book GiGi 180        //新增 Book 這項商品以及賣家 GiGi 賣此商品 180元
search Book                 //搜尋 Book 的所有賣家資料
```

```
buy Book                //購買 Book 這項商品
buy Book                //購買 Book 這項商品
search Notebook         //搜尋 Notebook 的所有賣家資料
sort                    //列出目前拍賣的所有商品名稱
buy CD                  //購買 CD 這項商品
sort                    //列出目前拍賣的所有商品
report                  //產生 report tables
```

Output File Format:

SearchTable.txt　將商品名稱以及此商品的所有賣家資料存進去，每搜尋一次就存放一次搜尋結果，並以分隔線隔開搜尋結果(「30 個-」)。

Example:

```
CD
Mary 280
----------
Book
GiGi 180
Jacky08 220
Jacky01 200
----------
Notebook doesn't exist!    //搜尋失敗訊息(此為 comment，實際 file 不會有此行)
```

BuyTable.txt　列出成功交易的商品名稱、賣家ID、商品價格，每筆資料一列，以此類推；若交易不成功則顯示商品名稱及錯誤訊息。

Example:

```
Book doesn't exist!      //購買失敗訊息(此為comment，實際file不會有此行)
Book GiGi 180
Book Jacky01 200
CD Mary 280
```

SortTable.txt　將拍賣的所有商品名稱列出，每項商品資料一列，以此類推。並且每一次sort 的結果以分隔線隔開搜尋結果(「30 個-」)。

Example:

```
Book
CD
TV
----------
Book
```

```
TV
---------------------------
```

LogTable.txt　　Database Access Log (實際檔案無comment 部分)。

Example:

```
insert 5          //insert 總指令個數
search 3 1        //search 總指令個數 搜尋失敗個數
buy 4 1           //buy 總指令個數 購買失敗個數
node_num 2        //binary search tree 的 node 個數
height 2          //binary search tree 的 height
```

其他注意事項：

1. Input：要求使用者輸入input file 的檔名。
   Output：產生相關 4 個 files。
2. Demo時，助教會使用測試檔測試結果。
3. 程式的基本 data structure 和程式架構可參考後面的APPENDIX。
4. 請確定您的程式可以在Dev-C++ / Code::Blocks 環境下編譯執行。
5. min heap的演算法有很多種，以課堂上教的演算法為主。
6. Binary Search Tree的delete方法，以左子樹最大者取代或是右子樹最小者取代皆可。

Submission:

Filename format:
學號_PE#.c
例如: M06455505_PE6.c

- In-class demo (12/13 Tue. 13:10-16:00)
- iLearn 2.0 submission (Due at 12/12 Mon. 23:59)

* 用自己的學號建立資料夾，並將 source code 放入資料夾，壓縮上傳 iLearn 2.0

Grading:

Correctness	50%
Program structure	20%
Comments	10%
Header block	5%
Variable dictionary	10%
Procedures and functions	5%

Special notice:
請勿抄襲別人程式(助教會當場進行測問、判定)，或是遲交作業，否則一律 0 分計算。請一律使用 C 語言來撰寫程式，且必須保證你的程式能夠再 Dev-C++ / Code::Blocks 軟體上成功編譯與執行，使用其他程式語言一律不予計分
請依照題目給的輸入格式，否則不計分

==========

APPENDIX: Data Type

```
typedef struct SELLER_TYPE
{
    char SID[30];
    int PRICE;
};

typedef struct ITEM_TYPE *ITEM_PTR;
typedef struct ITEM_TYPE
{
    char ITEM_NAME[30];
    SELLER_TYPE SELLER_HEAP[8];
    ITEM_PTR LCHILD;
    ITEM_PTR RCHILD;
};
```

主要程式架構：

```
Main
{
    while(not end of file)
    {
        Token =token = get next_token;
        switch(token)
        case “insert” : search BST for item name;
                        if( item name is not found )
                        {
                            Insert the new item into BST;
                        }
                            Insert the new seller data into min heap;
                        break;
        case “search” : search BST for item name;
                        if( item name is not found )
                        {
                            Write error message into search data file;
                        }
                        else
                        {
```

```
                    Write all seller data of the item into search data file;
                }
                break;
        case “buy” : search BST for item name;
                if(item name is not found)
                {
                    Write error message into buy data file;
                }
                else
                {
                    Take from the root of the min heap, and write the seller
                    data of the root into buy data file;
                    Delete the root of the min heap, and adjust the min heap;
                    If( there is no seller in the min heap)
                    {
                        Delete the item from BST;
                    }
                }
                break;
        case “sort”: inorder traversal BST and write item names according to a to z;
                break;
        case ”report”: report database information into 4 files;
                break;
    }
}
```