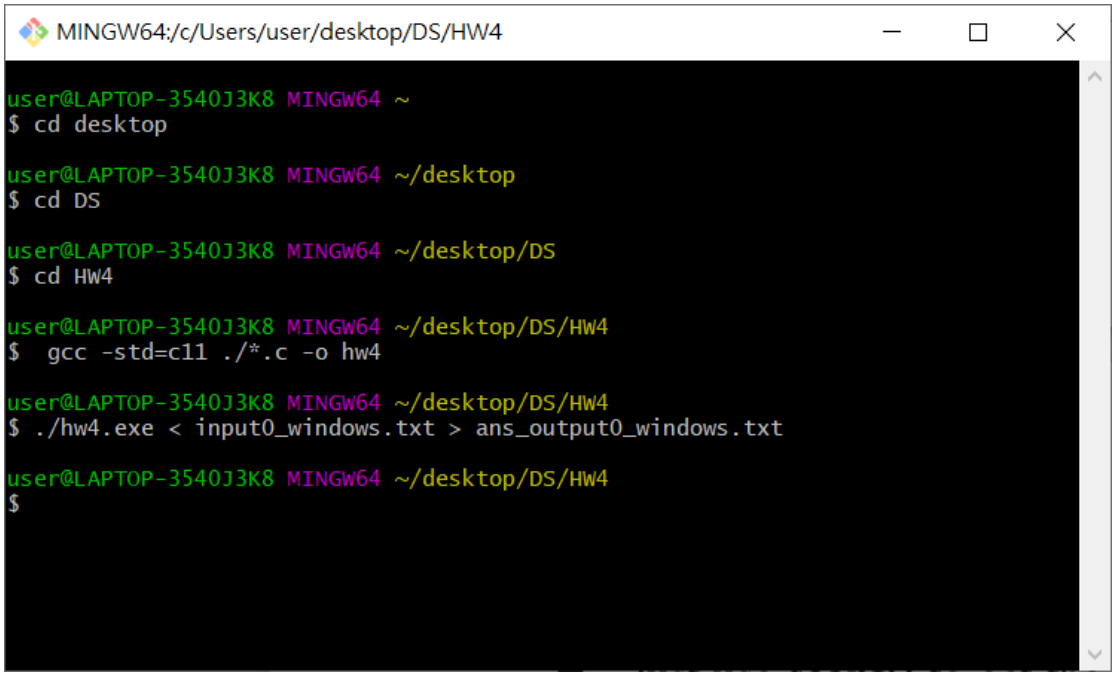
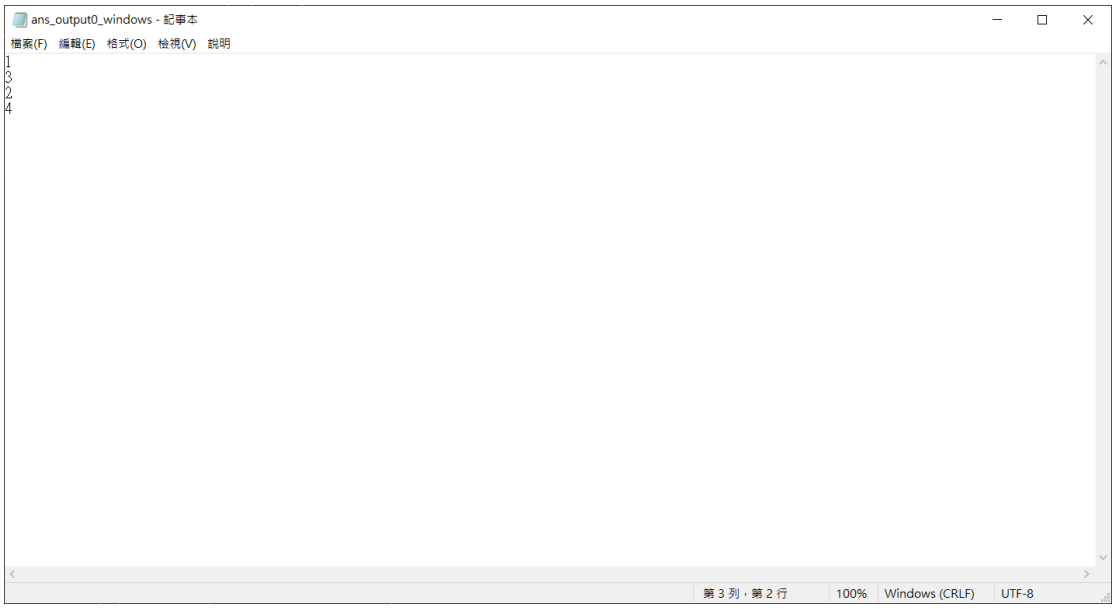
Result Screenshots

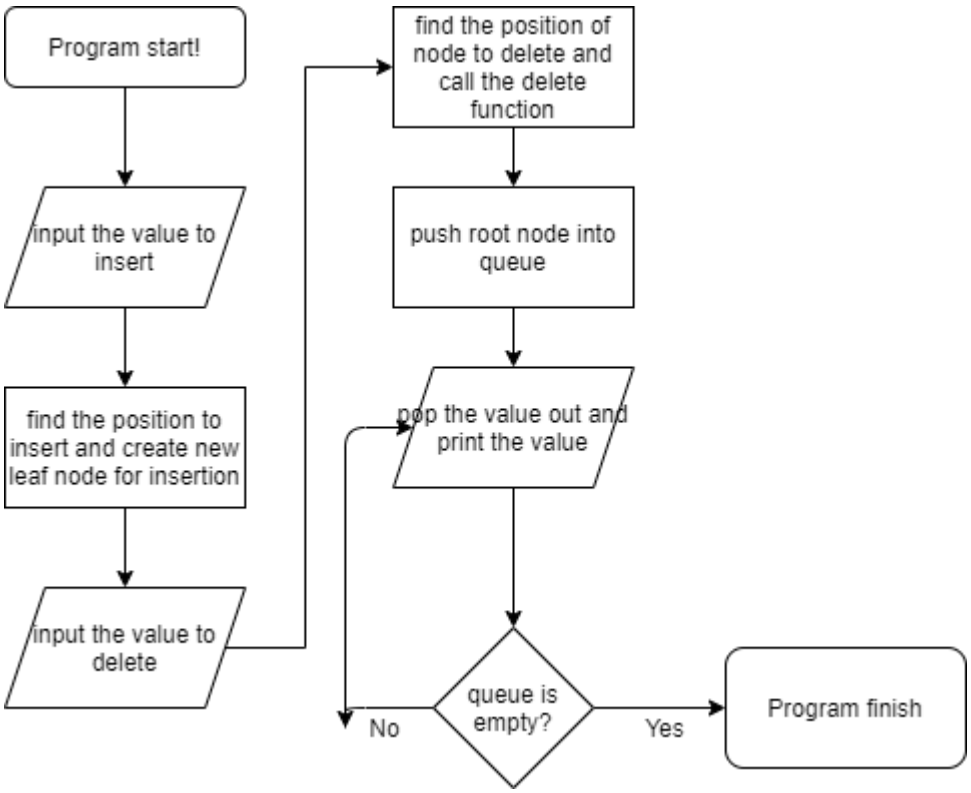


編譯與執行指令截圖



ans_output0_windows.txt 截圖

Program Architecture



Program Functions

node* find(int val);
//find the value's position in the binary search tree
Parameters:
int val: The value to find in the BST
Return value:
Return a pointer which points to the position of node where the value is found

void insert(int val);
//insert value into the binary search tree
Parameters:
int val: The value to insert in the BST
Return value:
No return value

void delete(int val);
// delete value from the binary search tree
Parameters:
int val: The value to delete in the BST
Return value:
No return value

void push(node* n);
//push value into queue
Parameters:
int val: The node pointer to push into queue
Return value:
No return value

node* pop();
//pop the first value out of queue
Parameters:
No parameters
Return value:
The node pointer which had been delete from queue

void printout();

//print the whole binary search tree by BFS

Parameters:

No parameters

Return value:

No return value

Program design

插入 BST: 對插入的數值做二元查找 找到對應的葉節點的位置 新增一個葉節點 並且把直設成插入的值

刪除: 先用二元查找 找到要被刪除的節點 如果是葉節點就直接刪除 如果有一個子節點 就把父節點與子節點串接後刪除 如果左柚子樹都不是空的 先找到左子樹的最大值 然後將左子樹最大值的節點移到被刪除的節點的位置 之後再重新串接此節點的子樹

Level order 的搜尋順序等價於廣度優先搜索(BSF) 用 queue+loop 跑到 queue 為空為止就完成了

Operating System

Windows10 家用版

Compiler

gcc version 8.2.0 (MinGW.org GCC-8.2.0-3)

Compile

gcc -std=c11 ./*.c -o hw4

Run

. ./hw4.exe < input0_windows.txt > ans_output0_windows.txt