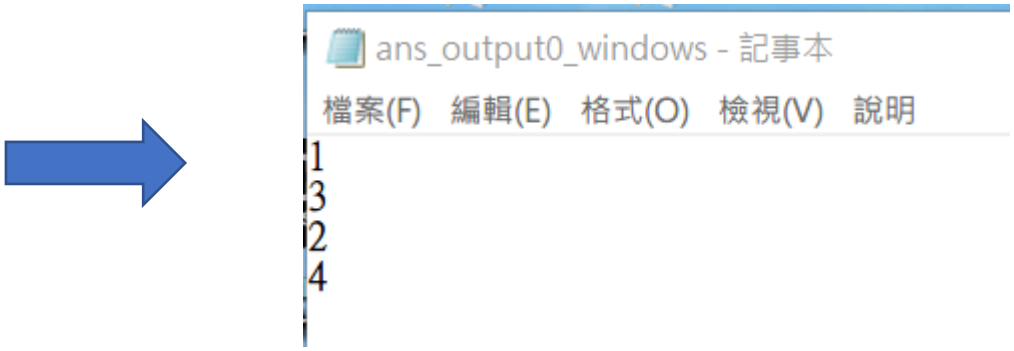
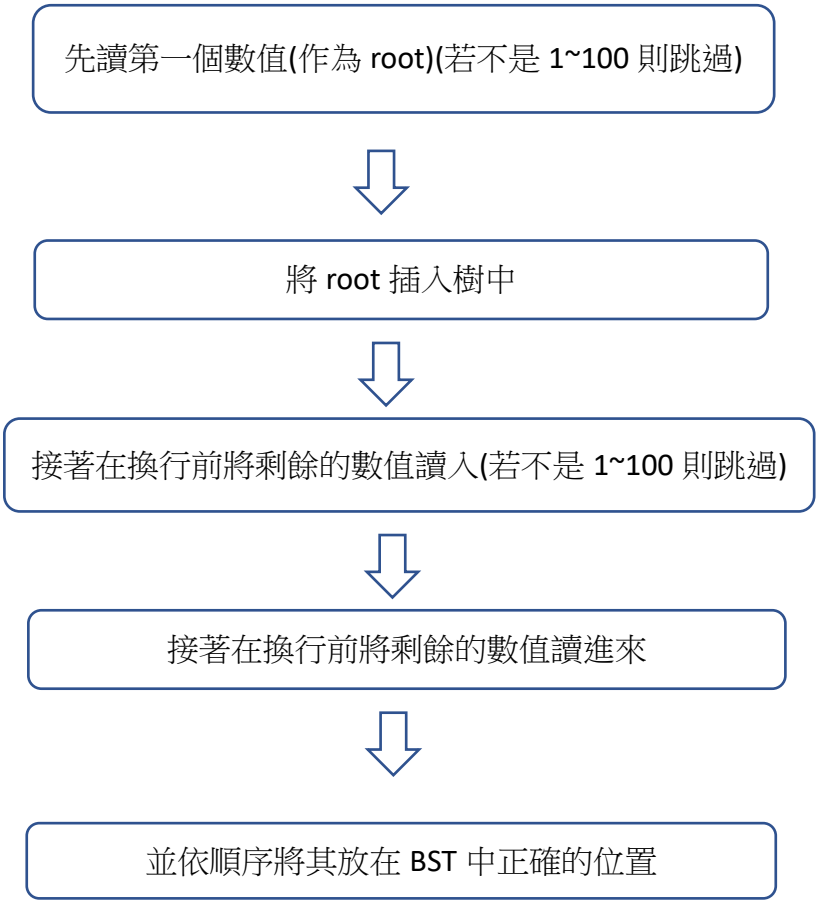
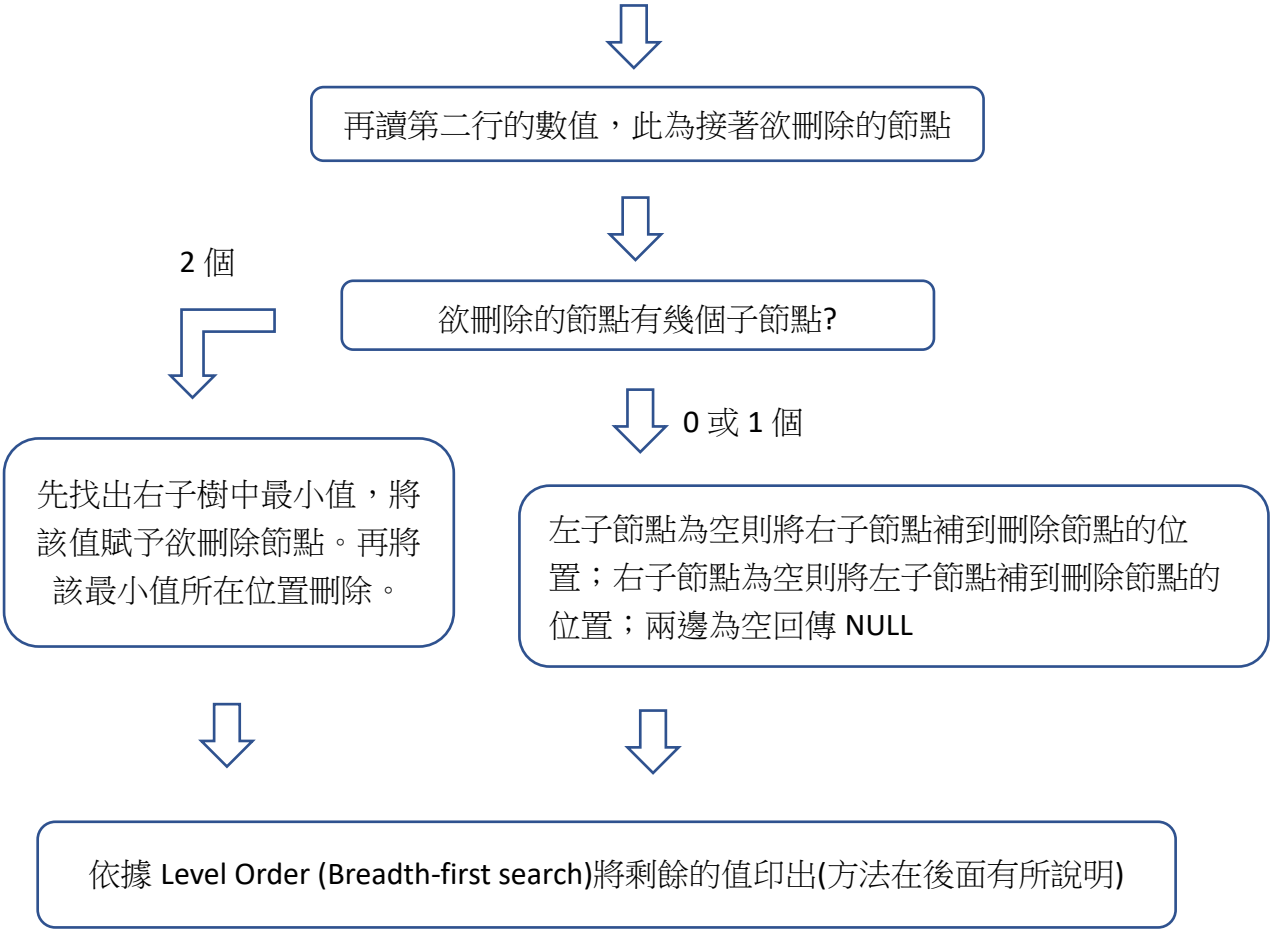
一、執行結果

```
User@LEO MINGW64 /d/Users/User/Desktop/DS_hw4/old (but correct)
$ gcc -std=c11 -o hw4 hw4.c

User@LEO MINGW64 /d/Users/User/Desktop/DS_hw4/old (but correct)
$ ./hw4.exe < input0_windows.txt > ans_output0_windows.txt

User@LEO MINGW64 /d/Users/User/Desktop/DS_hw4/old (but correct)
$ diff ./output0_windows.txt ./ans_output0_windows.txt

User@LEO MINGW64 /d/Users/User/Desktop/DS_hw4/old (but correct)
$
```



二、流程圖





三、函式說明

<自創函式>

struct node *newNode(int data)

- 當要 insert 的位置還沒有建立時(NULL)，將建立一個新的 node，其值為 data，其左右子節點預設為 NULL。完成後將此 node 回傳。

struct node *insert(struct node *node, int data)

- 當要 insert 的位置還沒有建立時(NULL)，先呼叫 newNode。否則開始比較，若 data 與 node 的值相同，表示數值重複，須回傳原 node 之值(代表沒有改變)；若 data 比 node 的值小，用遞迴的方式呼叫 insert，把傳入的 node 改為 node->left(往左找)；若 data 比 node 的值大，用遞迴的方式呼叫 insert，把傳入的 node 改為 node->right(往右找)

struct node *minNode(struct node *node)

- 先建立一個 tmp_node 指向 node，在 tmp_node 不為 NULL 且 tmp_node->left 也不為 NULL 的情況下，持續將 tim_node 指向它的 left。如此就能得到此樹中的最小值所在的 node(在最左邊)。結束後將 tmp_node 回傳。

struct node *delete (struct node *node, int data)

- 做刪除時，若 node 為 null 則回傳 null，若欲刪除之 data 小於目前 node 的值，則呼叫遞迴往左去找要刪的 node；若大於則往右尋找。當欲刪除之 data 等於目前 node 的值時，判斷此 node 有幾個子節點。在 0 個或 1 個的情況下，若左子節點為 null 就將右子節點回傳；右節點為 null 就將左子節點回傳。並將原 node 的記憶體空間釋出。若 node 有 2 個子節點，則先呼叫 minNode 找出右子樹中最小值所在位置。將該最小值賦值給 node 後，呼叫 delete 將擁有該最小值的 node 刪除。

void printLevelOrder(struct node *root)

- 這裡使用 queue 的概念來完成 Level Order Traversal。在當前 node 不等於 null 的情況下，將該 node 的值印出。接著若此 node 的左子節點存在，就將該左子節點放入 queue 中；右子節點同理。接著，只要 queue 不為空，就將當前節點變為 dequeue 出來的那個節點，繼續在 while 迴圈中執行。因為 queue 是 FIFO，所以 dequeue 出來的順序就是 enqueue 的順序，如此就能按照 level 一層一層往下印。

<引用函式>

char *strtok(char *str, const char *delim)

- 在第一次使用時，第一個參數要傳的是要分割的大字串，第二個參數是用於分割用的符號(看到該符號即要分割)。第二次以後的第一個參數則是傳 NULL，如此函式會從大字串剛才分割到的位置繼續開始分割，直到整個字串變為 null 為止。

四、設計要點

- Binary Search Tree: 是二元樹的一種。依照輸入值的順序第一個值設為 root，接著遇到比 root 小的就往左邊放，比 root 大的就往右邊放。使得隨便選一個節點，左子樹的值都會小於該節點的值，右子樹的值都大於該節點的值，且整棵樹的值都要是相異的。若有 n 個節點，則 BST 的空間複雜度是 O(n)，搜尋、插入和刪除的時間複雜度平均都是 O(logn)，最差則是 O(n) (每個節點都只有一個子節點的情況下)。