作答過程可參閱攜來之資料，唯須獨力完成，切勿與他人交換訊息或上網，違規者以零分計並立即請出場。欲繳卷的同學舉手表示作答完成，此時再恢復網路，把程式原始碼(.c或.cpp但不要.exe檔)或整個專案壓縮成單一檔案命名為final.zip再上傳至 *http://winlab.ee.yuntech.edu.tw/ds/* 教師當場確認收到繳交的內容之後，同學方可離席。學期成績將於下週二(1/17)16:00在課程網頁公佈，祝福大家都因著用心學習而高分過年關、寒假豐收!!

準備工作：請同學自 https://bit.ly/3w0vKdY 下載所需素材，妥善保存，之後將中斷網路。

1. [70 分] 本題只需底下(a)與(b)小題擇一完成。

(a) 擴充作業四程式碼，加入「引線」(thread)機制，成為「引線二元搜尋樹」(threaded binary search tree)。建構樹所需節點之資料型態定義為

```
typedef struct dict {          /* 引線二元搜尋樹節點之資料結構，實作字典 */
    char key[20];              /* 紀錄該節點的「鍵」(key)內容 */
    int  value;                /* 紀錄「鍵」對應之「值」(value)內容 */
    struct dict *llink;        /* 指向左子樹根或引線用途 */
    struct dict *rlink;        /* 指向右子樹根或引線用途 */
    int  lbit;                 /* 指示 llink 是否為正常指標 */
    int  rbit;                 /* 指示 rlink 是否為正常指標 */
} dict;
```

程式被限制不使用遞迴函數。程式的要求、輸入和輸出方式與作業四相同，完成的程式實例可參見下載素材之 threaded-dict.exe。未使用新訂 dict 資料型態完成要求者未能得分。

(b) 撰寫「引線二元搜尋樹」程式讀取一份檔案作為輸入，實作功能如下：

✓ 使用資料結構

```
typedef struct treeNode {      /* 引線二元搜尋樹節點之資料結構 */
    int  key;                  /* 紀錄該節點的「鍵」(key)內容 */
    struct treeNode *llink;    /* 指向左子樹根或引線用途 */
    struct treeNode *rlink;    /* 指向右子樹根或引線用途 */
    int  lbit;                 /* 指示 llink 是否為正常指標 */
    int  rbit;                 /* 指示 rlink 是否為正常指標 */
} node;
```

✓ 新增節點功能，亦可檢測元素有否存在樹內；若已存在，則忽略加入動作。

✓ 刪除節點功能，亦可檢測元素有否存在樹內；若不存在，則忽略刪除動作。

✓ 以非遞迴的中序追蹤(inorder traversal)方式走訪整棵樹。

✓ 讀取 bst.txt 檔案，內有增減節點資料與顯示樹內容的指令。譬如：“+50”指令意為新增鍵內容為 50 的節點至樹狀結構中；“?”顯示樹根資訊及中序追蹤整棵樹所得結果；“-50”意為刪除鍵內容為 50 的節點。

✓ bst.txt 檔案所載指令作為程式的輸入，毋須從鍵盤讀入額外內容；程式輸出畫面則依所載指令的要求顯示動作結果，包含動作“成功”或“捨棄”之狀態訊息。

程式的運行方式可參見下載素材之 threaded-bst.exe，部分程式碼請見 threaded-bst.c，請同學接手完成。未使用新訂 node 資料型態完成程式者未能得分。

註：相較於(a)小題，(b)小題不需實作作業四解答的 modify()與 deleteAll()函數，且節點之「鍵」欄位資料相對單純、無涉字串的比較，亦無「值」欄位，整個程式所需複雜度較低。

2. [30 分] 撰寫「橫向優先搜尋」(breadth-first search)程式，程式讀入 bfs.dat 檔案，該檔案紀錄一個相鄰矩陣(adjacency matrix)但以上三角形式呈現頂點之間的連接關係。程式的運行方式可參見下載素材之 bfs.exe。

3. [10 分，多選題] 下列敘述何者為真？

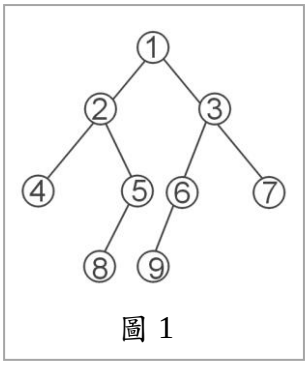

圖 1

(a) 回收 *n* 個節點的環狀串列(circular linked list)所需運算複雜度為 O(1)。
(b) 回收 *n* 個節點的雙向串列(doubly linked list)所需運算複雜度為 O(1)。
(c) 圖 1 所示是一棵二元樹，但不是滿枝二元樹(fully binary tree)，也不是完整二元樹(complete binary tree)。
(d) 圖 1 所示的二元樹之後序追蹤(postorder traversal)為 4, 8, 5, 2, 1, 9, 6, 7, 3。
(e) 任何一棵二元樹從根節點開始進行前序(preorder)追蹤與縱向優先搜尋(depth-first search)能得到相同的拜訪節點順序。
(f) 任何一棵二元樹從根節點開始進行後序追蹤與拓樸排序(topological sort)能得到相同的拜訪節點順序。

4. [10 分] 一棵二元樹之中序追蹤所得節點的拜訪順序為 DBACE，前序追蹤為 ABDCE，對應的二元樹為何，請繪製推演過程。

5. [10 分] 有一網路如圖 2 所示，以 Prim's 或 Kruskal's algorithms (擇一)求出最小成本擴展樹，請繪製求解過程。

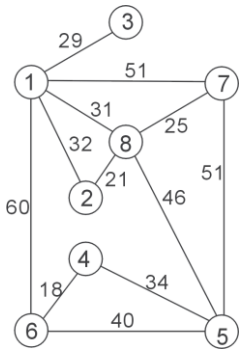

圖 2

6. [10 分] 有一網路如圖 3 所示，使用 Dijkstra's 演算法計算頂點①到頂點⑧的最短路徑為何，請明列求解過程。

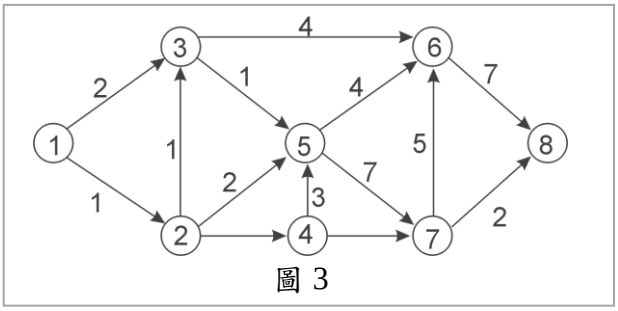

圖 3