DB HW4：B+Tree

【題目敘述】

B+Tree 可以在對數時間$O(log\ n)$ 內完成 Insert、Search、Delete也因為資料都存在 leaf 的特點，讓 B+Tree 可以有效的對資料進行序列性存取，因此 B+Tree 經常用於資料庫和作業系統的檔案系統中，本次作業將用 C 及 C++ 來實作 B+Tree 的 Insert、Search 及 Sequential data access。

【輸入說明】

第 1 行是一個正整數 M（且 M 為奇數，$M < 20$），表示 B+Tree 的 Order（每個 node 最多有 M 個 child），接下來每一行都會是以下其中一種指令：

`i x`	插入一個值為 `x`（`x` 為一個 integer 範圍的整數，且同一個測資中插入的值皆不同）
`s x`	在 B+Tree 中尋找 `x`（`x` 為一個 integer 範圍的整數）
`p`	印出整棵 B+Tree
`q`	結束程式（測資最後都會有這行）
`a x N`	印出 value 為 `x` 開始的 `N` 筆連續資料（`x` 及 `N` 皆為 integer 範圍的整數，`N > 0`）

【輸出說明】

請針對每個輸入的指令做出對應的輸出：

`i x`	不需要輸出
`s x`	請從 root 開始，依照印出所經過的每一個 node（包含 root） 每行印出一個 node 的所有 value，用空格隔開 value，並用 “`()`” 將該 node 包起來 若找到 `x`，請再輸出一行 “`Found`”，反之，請輸出 “`QAQ`”
`p`	**依照 pre-order 的順序**印出 B+Tree 中的每一個 node（包含 root） 每行印出一個 node 的所有 value，用空格隔開 value，並用 “`()`” 將該 node 包起來 另外，對於第 N 層的 node（root 在第 0 層），請在句首加上 $2N$ 個空白進行縮排
`q`	不需要輸出
`a x N`	若 `x` 不存在於 B+Tree 中，請輸出 “`Access Failed`” 若 `x` 存在於 B+Tree 中，請依序輸出由 `x` 開始的 `N` 個 value（包含 `x`） 請用空格隔開 value，另外根據 `N` 的大小會有兩種情況： (1) `x` 之後（含 `x`）的 value 有 `N` 個以上： 不用額外處理，請依照前面的敘述輸出即可 (2) `x` 之後（含 `x`）的 value 不足 `N` 個： 請將 `x` 之後的 value 都輸出，並在下一行輸出 “`N is too large`”

注意：在 `s`、`p`、`a` 輸出完之後，請再輸出一行空白行，來區分其他指令的輸出。

【範例 **1**】(無 Sequential access)

輸入：

```
3
p
i 10
i 90
i 50
p
s 30
s 50
i 20
i 40
i 30
p
q
```

輸出：

```
()

(50)
  (10)
  (50 90)

(50)
(10)
QAQ

(50)
(50 90)
Found

(30)
  (20)
    (10)
    (20)
  (50)
    (30 40)
    (50 90)
```

【範例 **2**】(有 Sequential access)

輸入：

```
5
i 50
i 55
i 60
i 65
i 70
i 75
i 80
i 85
i 90
i 45
i 40
i 35
i 30
i 25
i 20
i 15
i 10
p
a 87 4
a 55 4
a 55 10
q
```

輸出：

```
(60)
  (30 45)
    (10 15 20 25)
    (30 35 40)
    (45 50 55)
  (70 80)
    (60 65)
    (70 75)
    (80 85 90)

Access Failed

55 60 65 70

55 60 65 70 75 80 85 90
N is too large
```

【注意事項】

1. Deadline：**6/17 (四) 23:00**
2. HW4 為單人作業，請記得每個人都要繳交。
3. 請用 **C** 及 **C++** 來完成本次作業。
4. 本次作業不 Demo，改以寫 Report 的方式（請見下方**【Report】**說明）。
5. 請將所有檔案（report.pdf、.c、.cpp、.h、.....）壓縮成一個 .zip 檔，檔名請依照 **HW4_{student_id}.zip**（例：HW4_0512345.zip）上傳到 E3。
6. HW4 分數計算：測資分數 **(50%) + Report (50%)**
   （測資的部分大概會有一半是有 Sequential access）

【Report】（請轉成 PDF 檔，連同程式碼一起放進壓縮檔中）

1. (10%) 請寫出在 linux 環境中編譯及執行（各 5%）你的程式的指令，例如：

   ```
   $ gcc HW4.c -o HW4
   $ ./HW4
   ```

   如果有提供 Makefile 的話，只要寫 make 的指令以及產生的執行檔名稱即可。
   注意！請確保你的程式能夠用本題回答的指令來編譯及執行，
   否則測資分數可能會 0 分，本題也可能會 0 分，可以在學校的工作站上先做測試。

2. (5%) 請比較 B Tree 及 B+Tree 的不同。
   (5%) 以及這些不同可以讓 B+Tree 獲得什麼好處。

3. (10%) 請詳述本次作業中 B+Tree 的結構，即 B+Tree 中的 node 是用什麼資料結構，還有是如何存放 value 及 pointer。

4. (5%) 請詳述本次作業中是如何實作 B+Tree 的 Insert 功能。
   (5%) 如果 Insert 後的 node 需要進行分裂的話，是如何實作的。

5. (10%) 請截圖執行完**【範例 1】**及**【範例 2】**後的結果（各 5%）。

【Reference】

- https://www.db-book.com/db7/slides-dir/PDF-dir/ch14.pdf
- https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html