

Programming assignment #4

實作 Binary Search Tree

目標

1. 熟悉 BST

題目描述

實作一個 binary search tree 以及它的 insertion、deletion、search 功能。

1. **Insertion:** 輸入一個整數，你需要建一個新的節點、並以這個整數做為 BST 排序的 key 值。如果 BST 已經儲存過該 key 值，則不需要做任何事情。整數的範圍介於 -2^{31} 到 $2^{31}-1$ 之間。
2. **Deletion:** 輸入一個整數，你需要刪除 BST 中擁有此 key 值的節點。如果 BST 內沒有該 key 值的節點，則不需要做任何事情。如果此節點有兩個小孩，你必須選擇左子樹裡面 key 值最大的節點來取代被刪除的節點，如果你不是以這種方式來刪除的話，將不會得到分數。整數的範圍介於 -2^{31} 到 $2^{31}-1$ 之間。
3. **Search:** 輸入一個整數，你需要將 BST 中擁有此 key 值節點的 Level 寫入輸出檔案中。如果 BST 內沒有該 key 值的節點，則將 -1 寫入輸出檔案中。在這份作業中，我們定義 root 的 level 為 0。整數的範圍介於 -2^{31} 到 $2^{31}-1$ 之間。

此外，還有一個加分題的部分，為實作 search by rank 的功能，會輸入一個整數 k，你需要輸出在此 BST 中 key 值第 k 小的節點的"key 值以及 Level"，並將結果寫到輸出檔案中。若 k 大於 BST 的節點數量，則不需要做任何事情。k 的範圍介於 1 到 $2^{31}-1$ 之間。

輸入 / 輸出格式

必須以命令列參數(command line arguments)的形式來進行讀檔以及輸出，所以輸入及輸出的檔名不能是固定的。執行檔案的命令為：

`./a.out input_file_name output_file_name`

輸入檔案中，每一行的第一個數字代表要執行的指令：

1 : insertion

2 : deletion

3 : search

4 : search by rank

指令後面則是該指令對應的參數。

輸入檔案範例

```
1 40
1 60
1 20
1 10
3 10
1 30
1 50
2 20
3 10
4 4
```

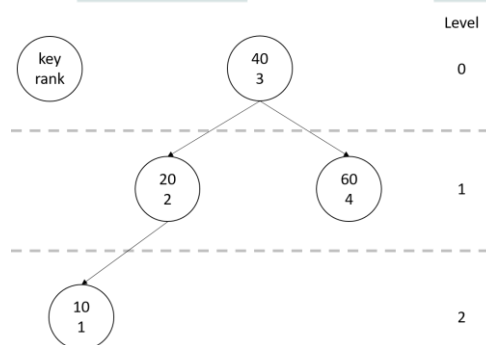


Fig. 1

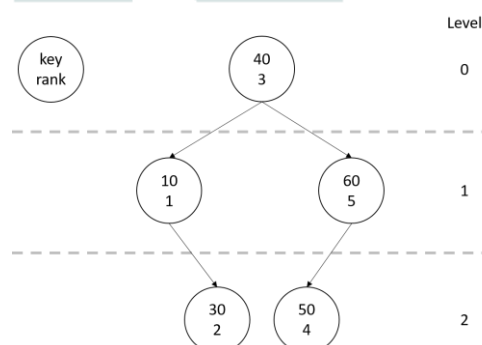


Fig. 2

輸出檔案需記錄所有指令 3、4 所應該輸出的值，其中所有數值必須以空格或 tab 或換行字符隔開。Demo 時並不是以 diff 這個指令進行比較的，所以間隔的字符不同並不會影響 demo 結果。

以上述輸入檔為範例：

當執行到第 5 行的"3 10" search 指令時，當下的 BST 如 Fig. 1 所示，擁有 key 值 10 的節點位於 Level 2，則輸出 2。

當執行到第 9 行的"3 10" search 指令時，當下的 BST 如 Fig. 2 所示，擁有 key 值 10 的節點位於 Level 1，則輸出 1。

當執行到第 10 行的"4 4" search by rank 指令時，當下的 BST 如 Fig. 2 所示，第 4 小的節點 key 值為 50、位於 Level 2，則輸出 50 2。

輸出檔案範例

```
2
1
50 2
```

作業繳交

1. 請用 C/C++ 來完成這份作業。
2. 檔名請命名為"ID_pa4.cpp"，例如：如果你的學號為0610101，你上傳的檔名必須為"0610101_pa4.cpp"。若因命名錯誤，造成demo問題，請於補交期間內補交正確的格式，以獲得補交分數。
3. 不論是以C還是C++來完成這份作業，都請將檔名命名為指定的格式，我們會以" g++ -std=c++11 *.cpp " 的指令來進行編譯。

配分

會有 10 個測資是以指令 1~3 所構成，若全部通過可以得 100 分。

有額外 2 個測資是以指令 1~4 所構成，若全部通過可以得額外的 20 分。

作業有開放補交，以相同的標準給分，但最後總分會再乘以 0.7。

每個測資的 runtime 不能超過 10 秒鐘。

繳交期限

請上傳檔案到 E3 平台

繳交期限為 **23:59 December 10**

補交的繳交期限為 **23:59 December 17**

