Programming assignment #3

Binary Search Tree

Objective

1. To understand how an implementation of an ADT is used by an application program.
2. To become familiar with how to implement binary search tree.

Problem Definition

給一顆二元搜尋樹(binary search tree)，將彈珠由根(root)丟下，每次往下掉落一層；彈珠會根據經過的節點(node)狀態決定往左子樹或往右子樹掉落，重複此行為直到下個掉落處沒有節點為止，你需要將第 N 顆球由 root 到結束所經過的所有節點依序輸出。

I/O Format

程式將會以下面的方式執行：

```
./a.out  input_file_name  output_file_name
```

Input 共有四種指令，分別為 add、deleteR、deleteL、drop。

1. add：依照 BST 規則，增加一新節點進入樹中，節點內容包含 id(無號整數)、初使方向(L 表示向左、R 表示向右)、改變方向條件(以一非負整數 X 表示，0 代表永遠不換、其他則表示需被 X 顆彈珠經過後才換方向)；另外，若是欲加入之節點 id 已存在，則將新的資料取代原節點中的內容。
2. deleteR：刪除指定 id 的節點，若是該節點有兩個 child，則以右子樹的最小值那個節點取代被刪除之節點。
3. deleteL：刪除指定 id 的節點，若是該節點有兩個 child，則以左子樹的最大值那個節點取代被刪除之節點。
4. drop：給一整數 N (N>0)，問依目前樹的狀態，丟下第 N 顆球時，球經過的路徑為何，並將該樹的狀態更新為丟完 N 顆球後的狀態。

Output 部分，針對 input 的每個 drop 指令，都需輸出一行以 id 表示的路徑，該路徑由 root 開始，將所有會經過的節點 id 輸出，每個 id 間請以一格空白隔開；若是樹中沒有任何節點，卻遇到 drop 指令，請輸出：Tree is empty.；除此之外，當遇到 deleteR、deleteL 指令，且欲刪除之節點 id 不存在時，需要輸出一行：Deletion failed. 以示警告。

Input file example (“//”符號為註解，不會在測資裡出現)

```
add 6 L 1      //add [id] [初始方向] [切換條件]
add 3 L 2
add 5 L 1
add 8 L 0
add 9 L 1
add 1 L 1
drop 3
deleteR 7
add 6 L 3
deleteL 6
drop 1
```

Output file example

```
6 3 1
Deletion failed.
5 3
```

Program Submission

1. Please use C/C++ language and your program **must** be written in **only one** source file.
2. Your source file must be named as “Student_ID_number_pa3.cpp” and please make sure that all characters of the filename are in **lower case**. For example, if your student number is 0510101, the name of your program file should be “0510101_pa3.cpp”.

Grading

You only need to submit your source code. Remember the submission rules mentioned above, or you will be punished on your grades.

- Unique and compilable source code 30 %
- Seven cases 70 %

Due Date

Upload your program to the e3 platform.

The upload deadline is **23:59 December 4**.

```
add 6 L 1
add 3 L 2
add 5 L 1
add 8 L 0
add 9 L 1
add 1 L 1
drop 3
deleteR 7
add 6 L 3
deleteL 6
drop 1
```

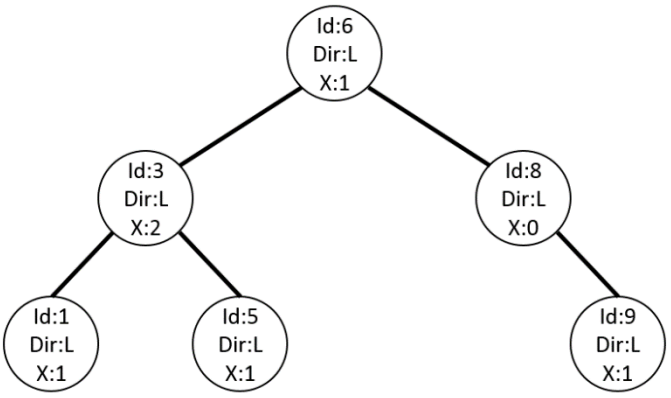

Fig. 1 初始樹

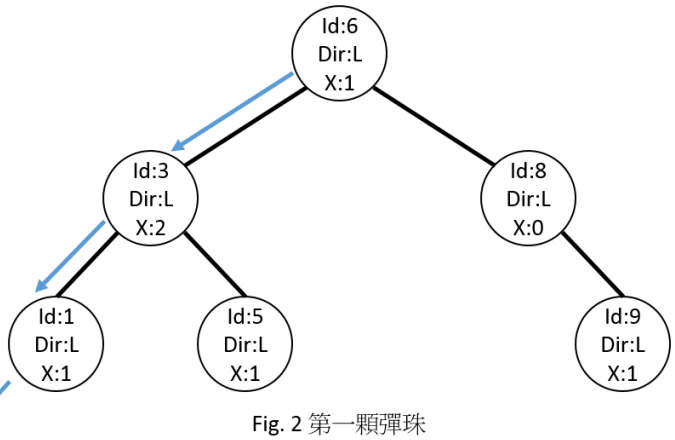

Fig. 2 第一顆彈珠

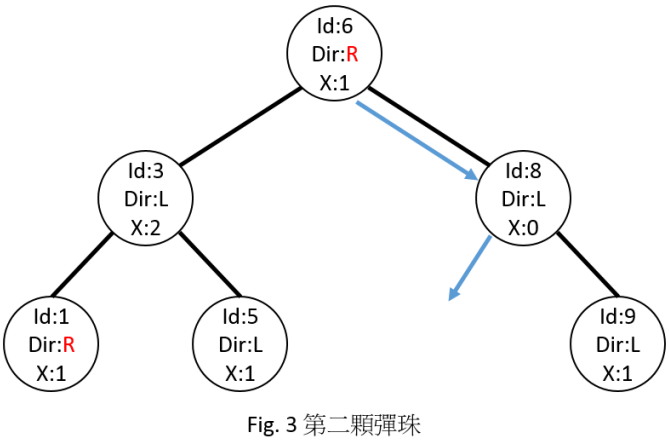

Fig. 3 第二顆彈珠

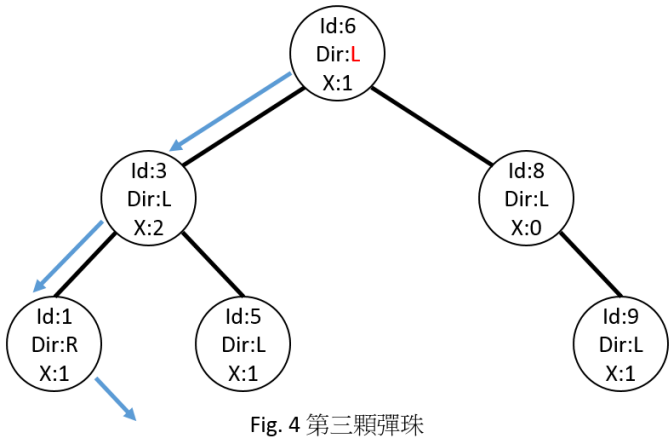

Fig. 4 第三顆彈珠

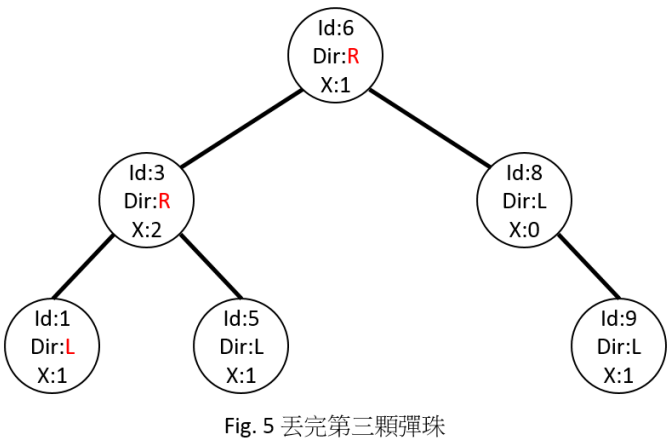

Fig. 5 丟完第三顆彈珠

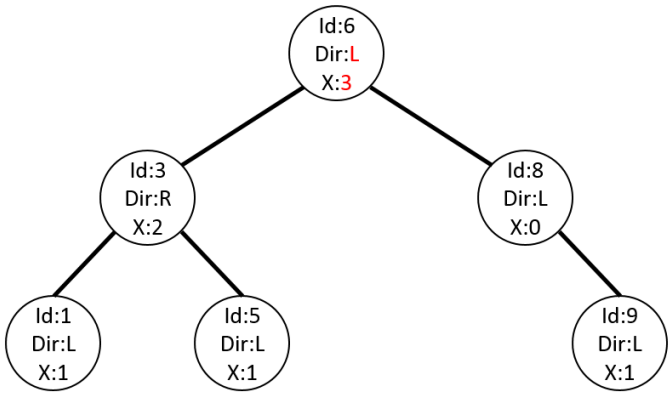

Fig. 6 加入節點 6 L 3

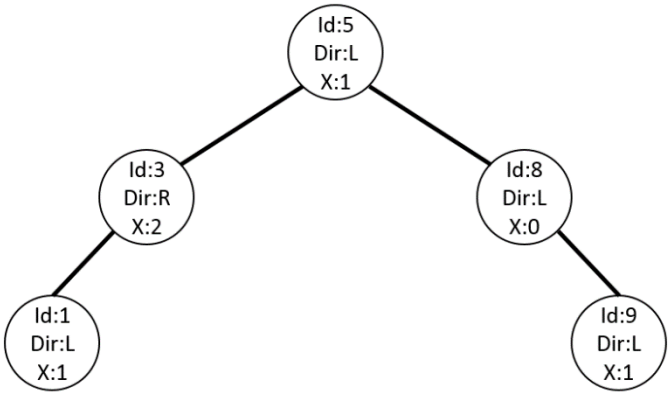

Fig. 7 deleteL 6

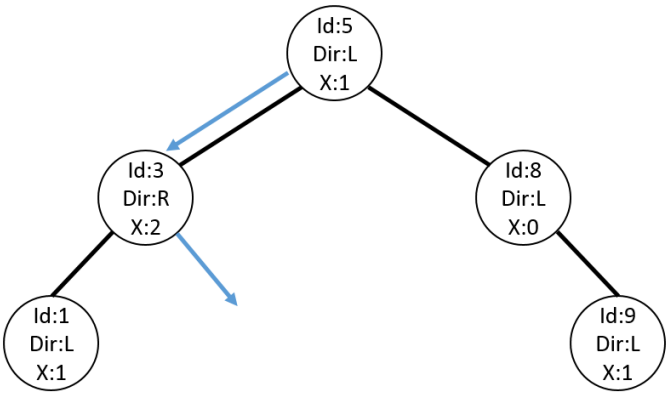

Fig. 8 再丟一顆彈珠