HW2 Binary Search Tree

Objective:

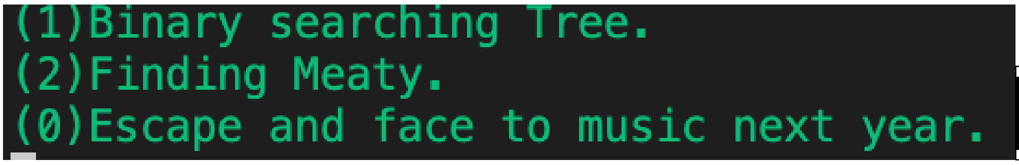
1. Implement the binary search tree (BST) operations **with linked list.**
2. Review of stack and queue.

Descriptions:

The HW2 is separated into two parts, i.e.

1. The first part is going to implement the operations of BST with linked list.
2. The second part “Finding Meaty” is going to make good use of several skills taught before.

The **main menu** of the program is shown as **Fig. 1**.

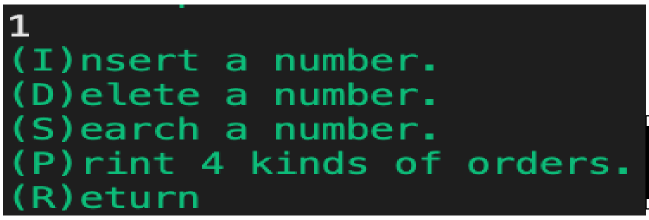


```
(1)Binary searching Tree.
(2)Finding Meaty.
(0)Escape and face to music next year.
```

Fig1

(60%) Part 1: BST operations

(10%) Output the **action menu** (as **Fig. 2**), and **don't forget a loop** for sustainably executing the actions.



```
1
(I)nsert a number.
(D)elete a number.
(S)earch a number.
(P)rint 4 kinds of orders.
(R)eturn
```

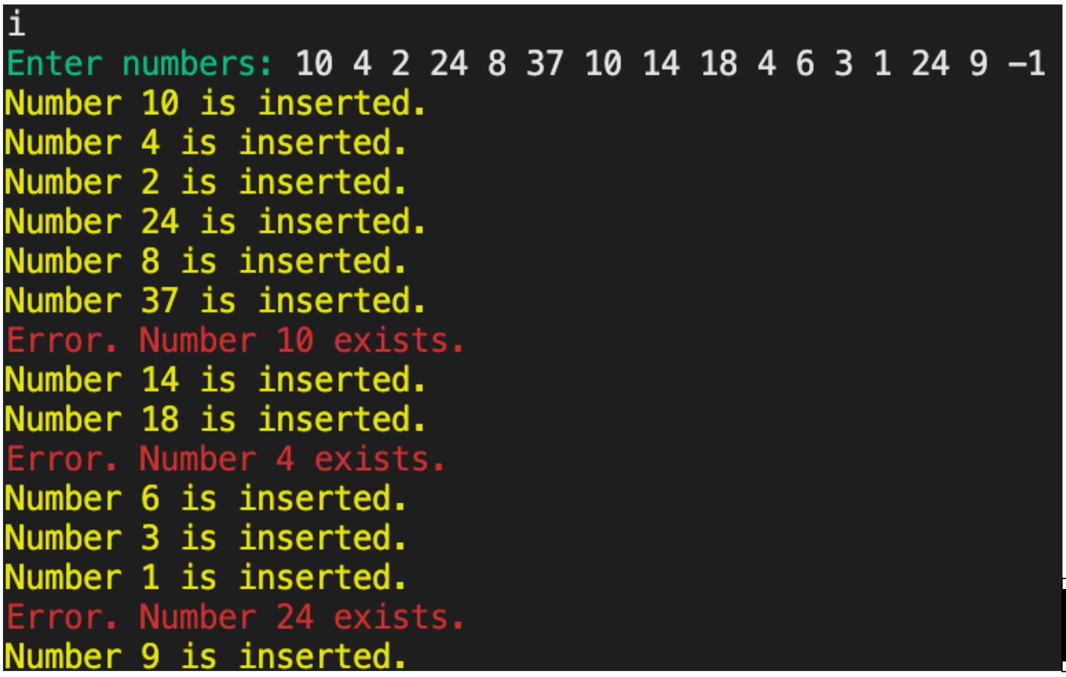
Fig2

(35%) Construct the BST.

1. Insert

Enter "i", then enter a sequence of numbers and ended with "-1" to build a BST. Noticed that the number **"-1"** is not a node for insertion.

Show the result as **Fig. 3** and also consider that these numbers have been inserted already.



```
i
Enter numbers: 10 4 2 24 8 37 10 14 18 4 6 3 1 24 9 -1
Number 10 is inserted.
Number 4 is inserted.
Number 2 is inserted.
Number 24 is inserted.
Number 8 is inserted.
Number 37 is inserted.
Error. Number 10 exists.
Number 14 is inserted.
Number 18 is inserted.
Error. Number 4 exists.
Number 6 is inserted.
Number 3 is inserted.
Number 1 is inserted.
Error. Number 24 exists.
Number 9 is inserted.
```

Fig3

2. Delete

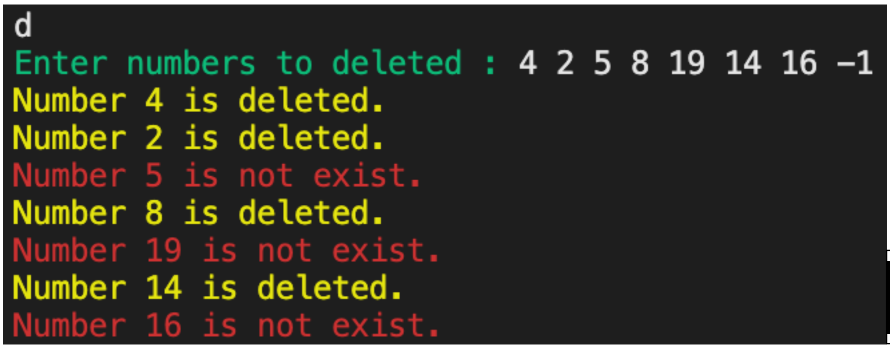
Enter “d”, then enter a sequence of numbers ended with “-1” to delete (the same as insert) as **Fig. 4.**

(a) If the deleted node is a leaf, then delete directly.

(b) If the deleted node has a child, then replace the deleted node with its child.

(c) If the deleted node has left and right sub-tree, then replace the deleted node with the smallest node of the right sub-tree.

Also consider the error deletion case that the deleted node(number) is not existed.



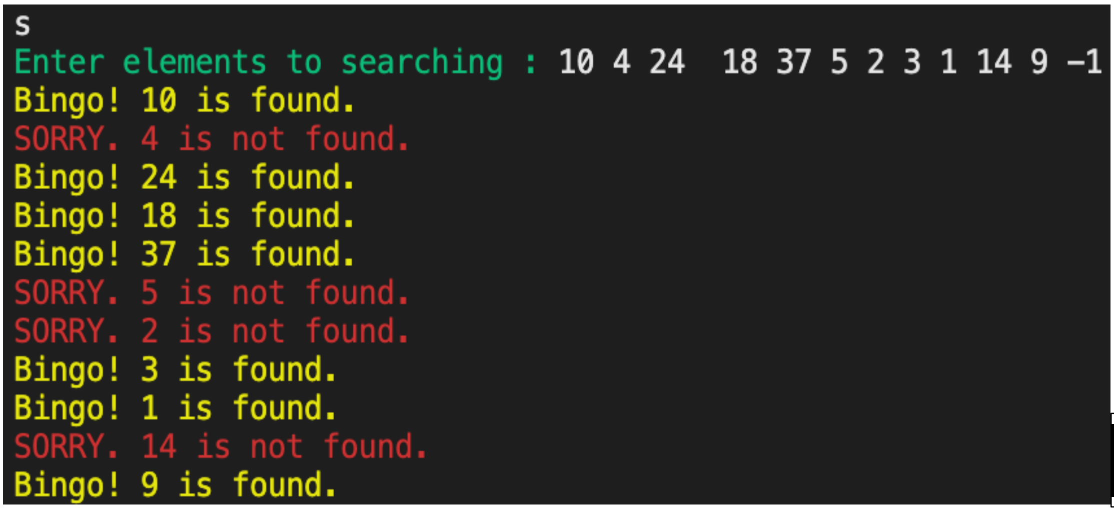
```
d
Enter numbers to deleted : 4 2 5 8 19 14 16 -1
Number 4 is deleted.
Number 2 is deleted.
Number 5 is not exist.
Number 8 is deleted.
Number 19 is not exist.
Number 14 is deleted.
Number 16 is not exist.
```

Fig4

3. Search

Enter “s”, then enter a sequence ended with “-1” to search whether the numbers are in the BST.

If the number is not in the BST then print the number is not found.



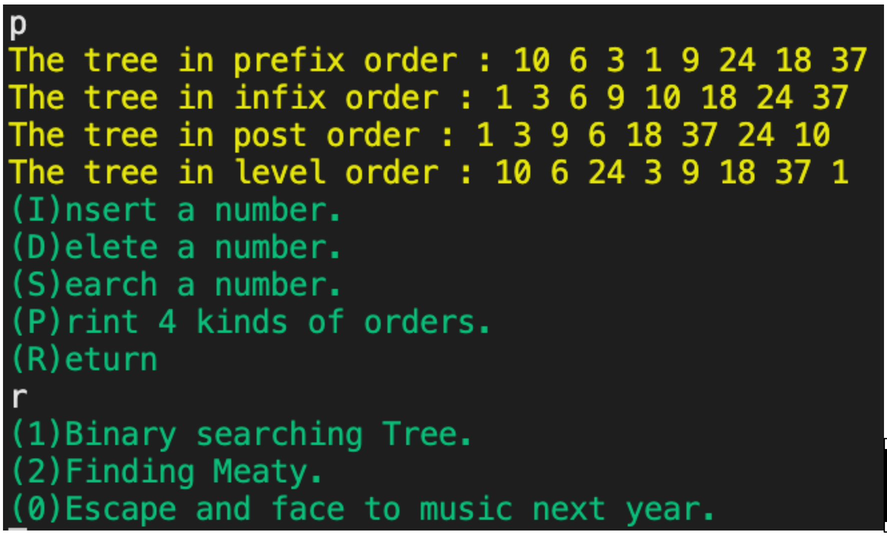
```
s
Enter elements to searching : 10 4 24  18 37 5 2 3 1 14 9 -1
Bingo! 10 is found.
SORRY. 4 is not found.
Bingo! 24 is found.
Bingo! 18 is found.
Bingo! 37 is found.
SORRY. 5 is not found.
SORRY. 2 is not found.
Bingo! 3 is found.
Bingo! 1 is found.
SORRY. 14 is not found.
Bingo! 9 is found.
```

Fig5

(15%) Print and return

Enter “p” to print prefix -> infix -> postfix and level order of the BST after the above executions.
Enter “r” to return to the main menu.

You must use queue skills in level order!



```
p
The tree in prefix order : 10 6 3 1 9 24 18 37
The tree in infix order : 1 3 6 9 10 18 24 37
The tree in post order : 1 3 9 6 18 37 24 10
The tree in level order : 10 6 24 3 9 18 37 1
(I)nsert a number.
(D)elete a number.
(S)earch a number.
(P)rint 4 kinds of orders.
(R)eturn
r
(1)Binary searching Tree.
(2)Finding Meaty.
(0)Escape and face to music next year.
```

Fig6

***Every execution above except [return] should be able to operate for several times.**

For example: insert more nodes after delete some nodes, and vice versa.

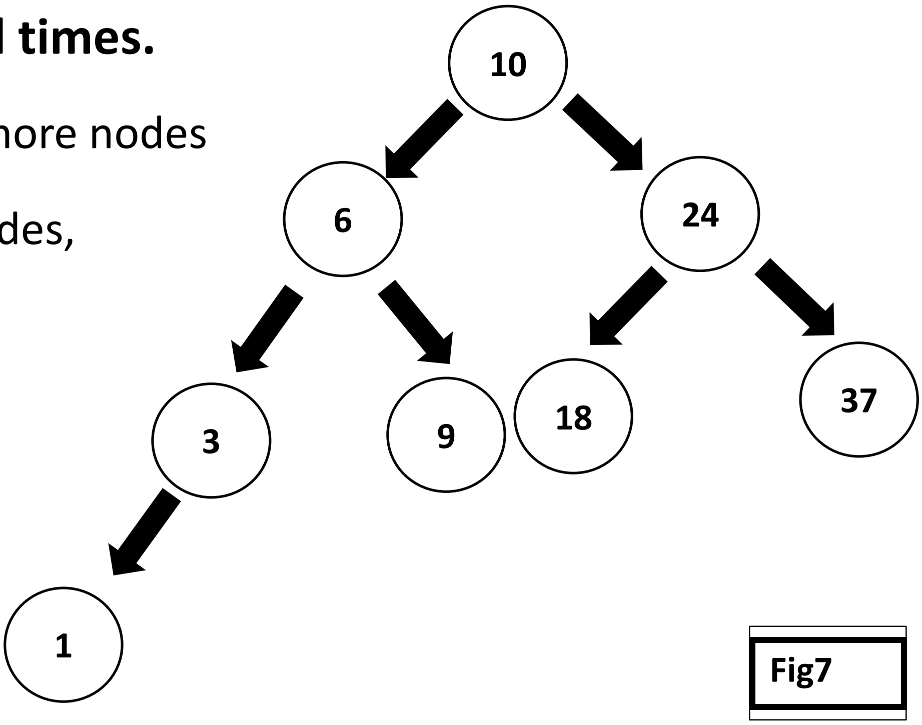


Fig7

(30%) Part2: Finding meaty

Capoo is going to find his favorite Meaty, but Meaty is caught by a dragon. Capoo must get a legendary sword first so that he can defeat the dragon and save Meaty.

However, the dragon set some trap on the map and you should help Capoo to dodge the broccoli first and then indicate him to get the sword and defeat the dragon in the fastest way.

An input file "bstmap.txt" is given and you should construct the map as a BST. The start position is always the root of the BST. Input three numbers:
(1) sword position
(2) Meaty(dragon) position.
(3) broccoli traps index.

The broccoli traps index is that **all of the nodes containing the index number are traps**.
For example: if the index number is 4, then 4, 14, 40 will all be traps.

Fig. 8 shows the BST build from the given “bstmap.txt” (**use variable filename when you read the file**), the sword position is 6, meaty position is 23, and the broccoli traps index is 4.

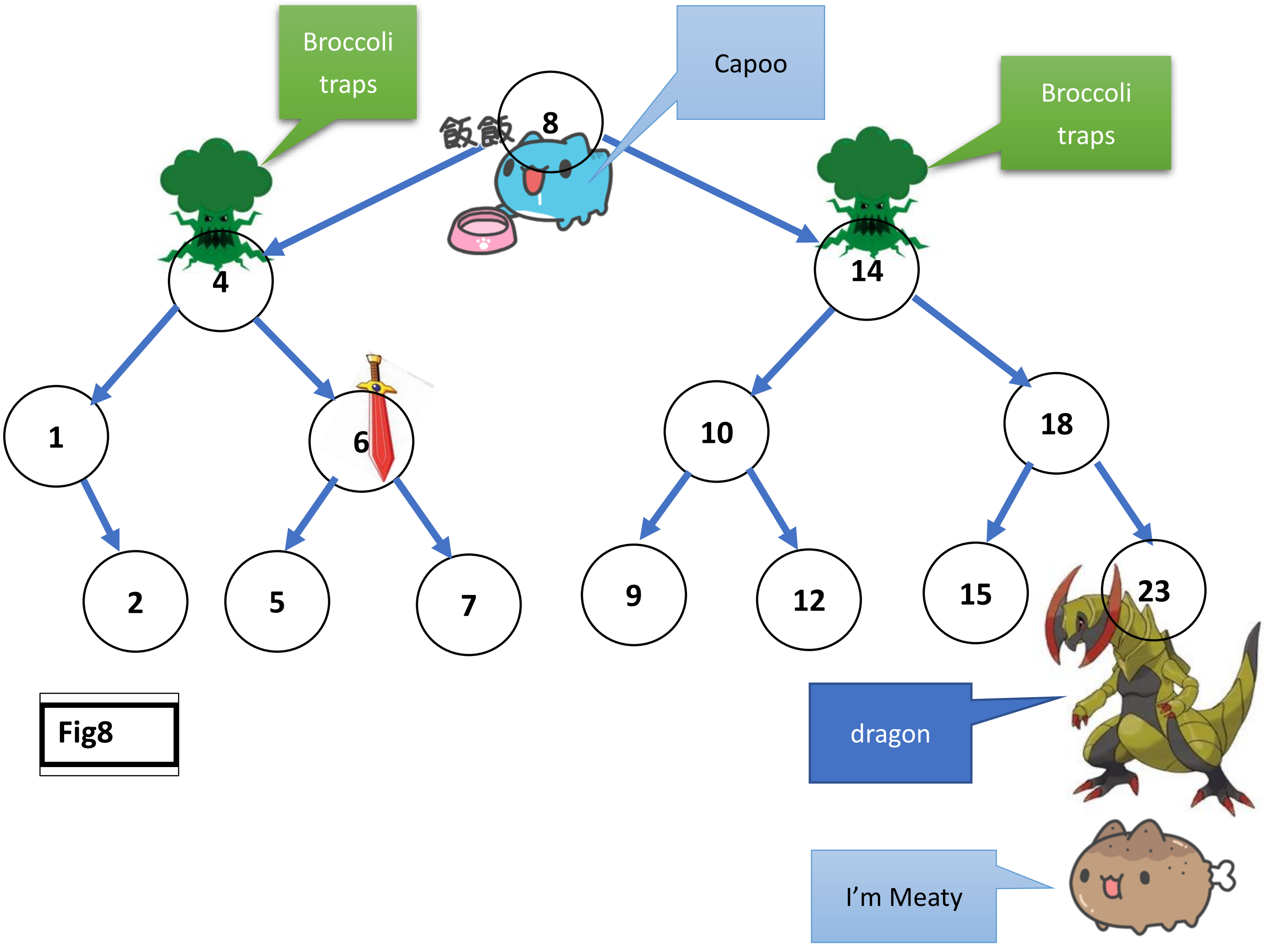


Fig8

You need to delete 4 and 14 first because they contain the broccoli traps index 4.

Then follow the same rules (a)(b)(c) in Part 1, replace the node 4 with node 5 and replace the node 14 with node 15, which are marked as orange nodes in **Fig. 9**.

The safe map is shown as the tree in **Fig. 9**.

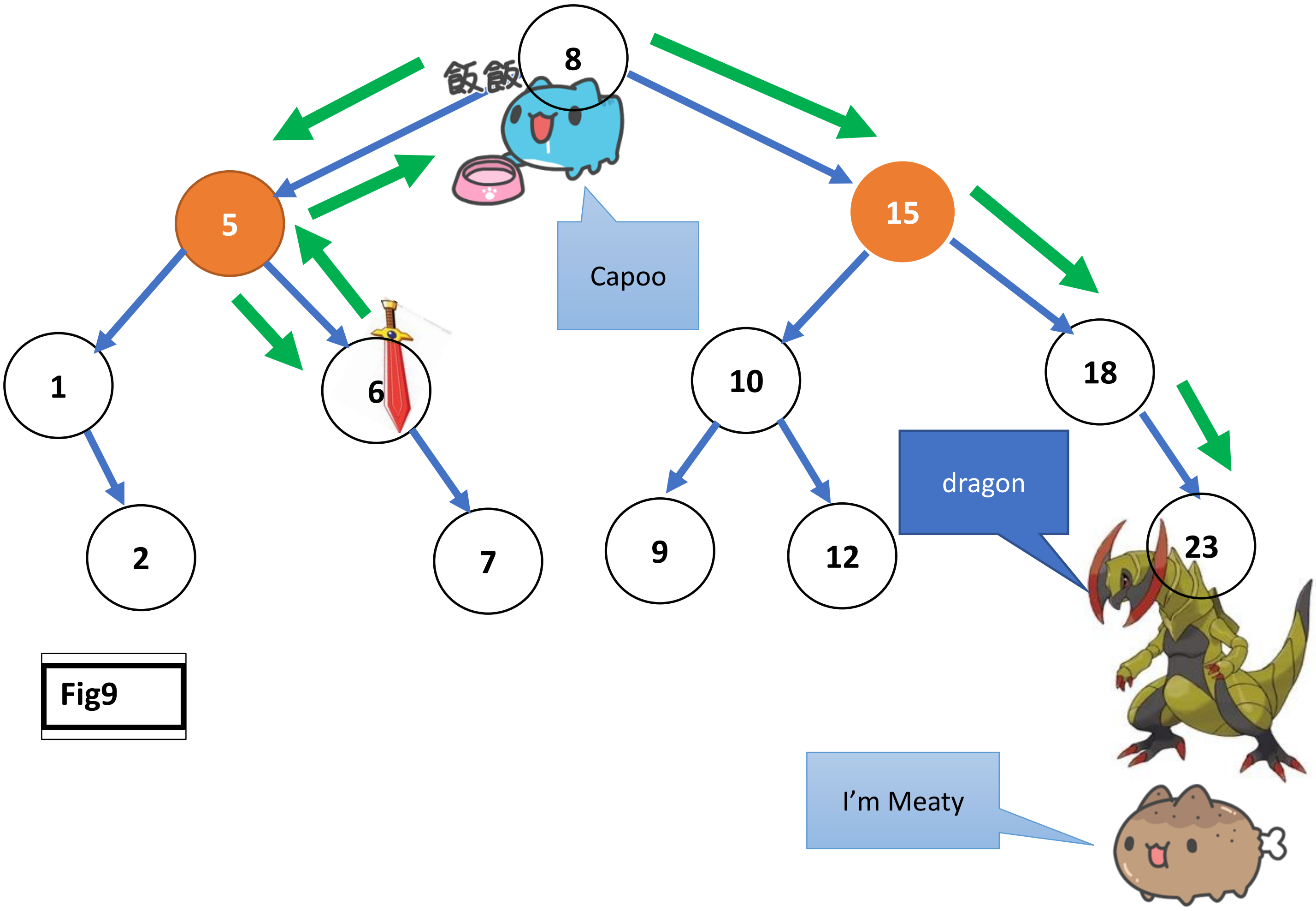
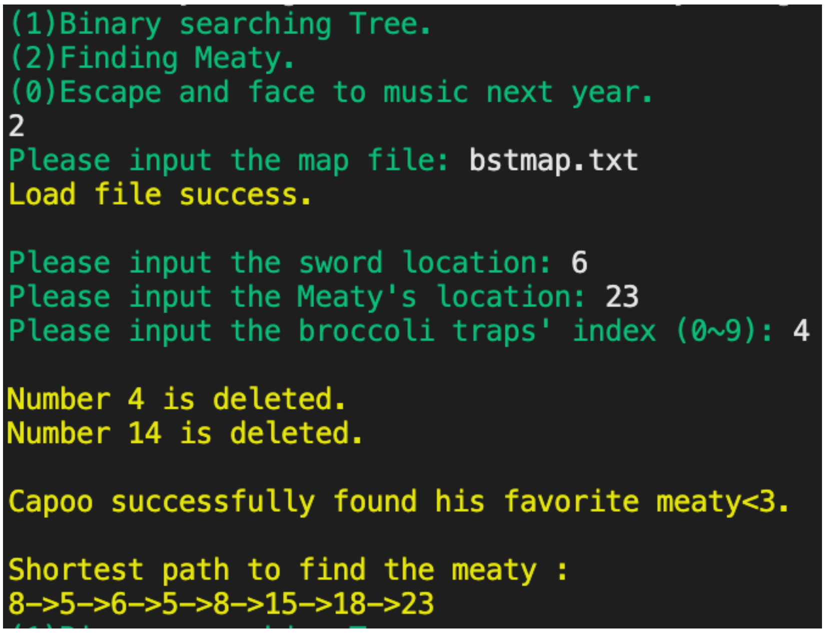


Fig9

Now Capoo can start his journey. He should visit node 6 first to get the sword and then go to the node 23 to challenge the dragon and save Meaty.

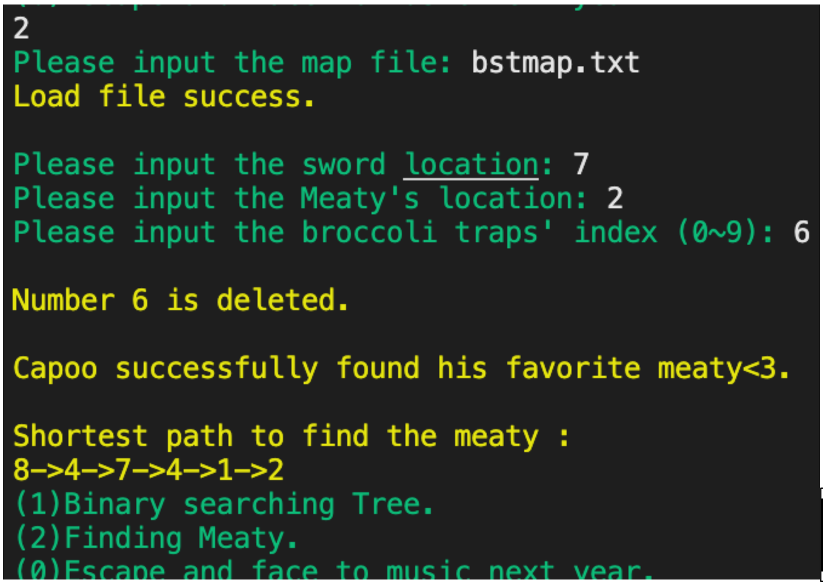
So the traversal of Capoo is 8->5->6->5->8->15->18->23, shown as the green arrows in Fig. 9 →

The result example:



```
(1)Binary searching Tree.
(2)Finding Meaty.
(0)Escape and face to music next year.
2
Please input the map file: bstmap.txt
Load file success.

Please input the sword location: 6
Please input the Meaty's location: 23
Please input the broccoli traps' index (0~9): 4

Number 4 is deleted.
Number 14 is deleted.

Capoo successfully found his favorite meaty<3.

Shortest path to find the meaty :
8->5->6->5->8->15->18->23
```

Fig10



```
2
Please input the map file: bstmap.txt
Load file success.

Please input the sword location: 7
Please input the Meaty's location: 2
Please input the broccoli traps' index (0~9): 6

Number 6 is deleted.

Capoo successfully found his favorite meaty<3.

Shortest path to find the meaty :
8->4->7->4->1->2
(1)Binary searching Tree.
(2)Finding Meaty.
(0)Escape and face to music next year.
```

Fig11

<u>Return to the main menu</u> after every execution of Finding Meaty, as shown at the bottom (last 3 lines) of **Fig. 11**.

(10%) delete the broccoli traps and replace with correct nodes

(20%) output the correct path of Finding Meaty

You must <u>use stack</u> to deal with this task.

(10%) readme file and comments

Comments should be in **English** otherwise it may be garbage code. Readme file can be in **Chinese** and should be stored as a **pdf** file.

Deadline:

Regular: at 11:59 p.m., Friday Dec. 17, 2021.
Note: Handing late will lead to a perceptibly lower score.

Hand in:

Zip your bst.cpp, readme.pdf to StudentID_hw2.zip and upload the zip file to Moodle.
The score will be deducted if the filename is not named with "bst.cpp", please DO follow the rule.

If you have any questions about the Assignment 2, you can e-mail to the TA:
Eric: p76101039@gs.ncku.edu.tw

*Discussion is encouraged, but DO NOT share your code to your classmate. TA will check the plagiarism issue. If plagiarism is found, both students will get 0 score in this assignment.