

Data Structure

Second Hands-on Test

Teacher: Mu-Yen Chen

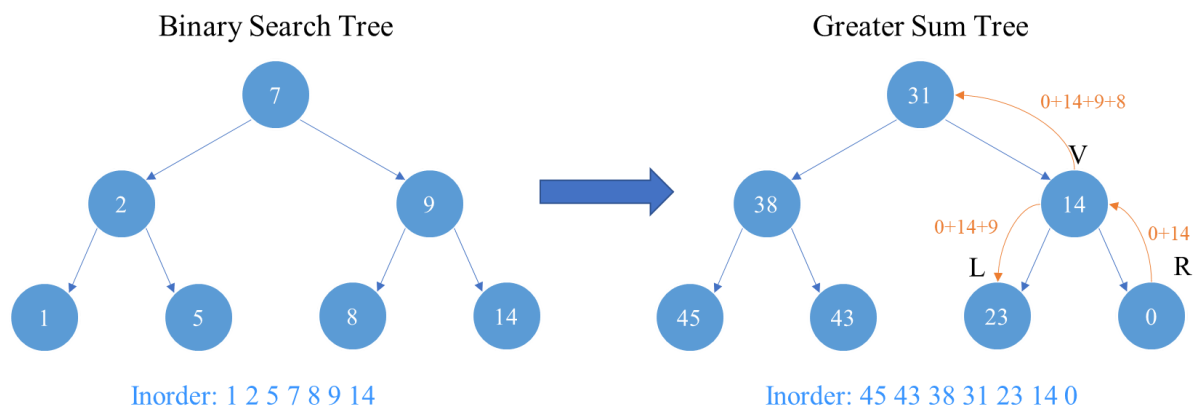
Date: Dec.1,2022

1. (50%)

Please write a code to construct a binary search tree (BST), it also has a function that can convert a BST into a **greater sum tree**. The greater sum tree is a tree in which every node contains the sum of all the nodes which are greater than the node. See the example below.

You can use P1.txt to test your code.

[Hint] Using RVL traversal to sum up.



```
1 7
1 2
1 1
1 5
1 9
1 14
1 8
1 15
2 15
4
3
4
```

P1.txt

```
The instruction:1. Insert 7. Insert successfully.
The instruction:1. Insert 2. Insert successfully.
The instruction:1. Insert 1. Insert successfully.
The instruction:1. Insert 5. Insert successfully.
The instruction:1. Insert 9. Insert successfully.
The instruction:1. Insert 14. Insert successfully.
The instruction:1. Insert 8. Insert successfully.
The instruction:1. Insert 15. Insert successfully.
The instruction:2. Delete 15. Delete successfully.
The instruction:4. Inorder: 1 2 5 7 8 9 14
The instruction:3. BST to greater sum tree.
The instruction:4. Inorder: 45 43 38 31 23 14 0
```

Output example

- (a) Insert data completely.(5%)
- (b) Delete data completely.(5%)
- (c) BST to greater sum tree completely.(35%)
- (d) In-order traversal completely.(5%)

2. (20%)

Given a directive graph which must have a topological sequence, please write a code to get its topological sequence.

```

7
0 1 1 1 1 1 1
0 0 1 1 1 1 1
0 0 0 1 1 1 1
0 0 0 0 1 1 1
0 0 0 0 0 1 1
0 0 0 0 0 0 1
0 0 0 0 0 0 0

```

P2.txt

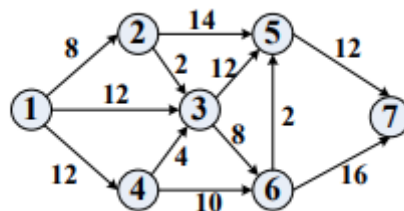
```

topological sort: 1 2 3 4 5 6 7
-----
Process exited after 0.03999 seconds with return value 0
請按任意鍵繼續 . . .

```

3. (30%)

Please modify the Dijkstra algorithm code from the fourth TA class, print the shortest path from the beginning to each node. (p.s. It's allowed to print the path backward.)



```

from 1 to 2      Shortest Distance = 8 path : 2 <- 1
from 1 to 3      Shortest Distance = 10 path : 3 <- 2 <- 1
from 1 to 4      Shortest Distance = 12 path : 4 <- 1
from 1 to 5      Shortest Distance = 20 path : 5 <- 6 <- 3 <- 2 <- 1
from 1 to 6      Shortest Distance = 18 path : 6 <- 3 <- 2 <- 1
from 1 to 7      Shortest Distance = 32 path : 7 <- 5 <- 6 <- 3 <- 2 <- 1

```

```

-----
Process exited after 0.03476 seconds with return value 0
請按任意鍵繼續 . . .

```