Binary Search Tree

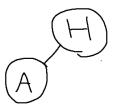
1. จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบ คำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

```
0. BST tree;
1. tree.insert('H');
2. tree.insert('A');
3. tree.insert('R');
4. tree.insert('H');
5. tree.insert('U');
6. tree.insert('I');
```

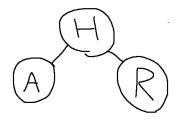
1.



2.



3.



Binary Search Tree

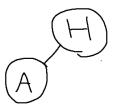
1. จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบ คำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

```
0. BST tree;
1. tree.insert('H');
2. tree.insert('A');
3. tree.insert('R');
4. tree.insert('H');
5. tree.insert('U');
6. tree.insert('I');
```

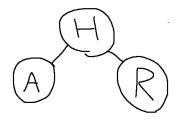
1.

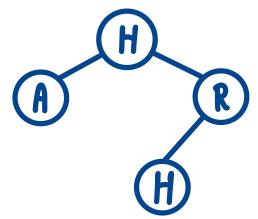


2.

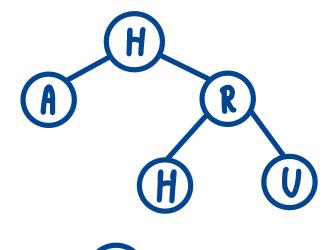


3.

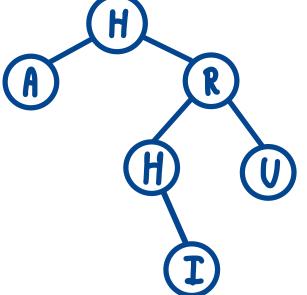




5.



6.



หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น ..H.A.R.H.I.V.

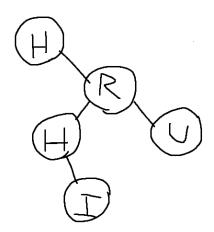
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น A H H I R U.....

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น ..AIHURII

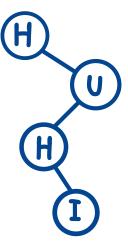
2. ต่อจากข้อ 1 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ใน โปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

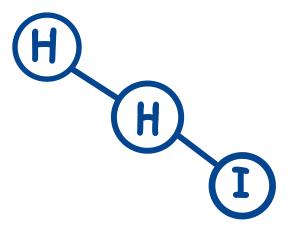
```
7.delete_node(&(tree.root->left));// A
8.delete_node(&(tree.root->right));
9.delete_node(&(tree.root->right));
```

7.



8.

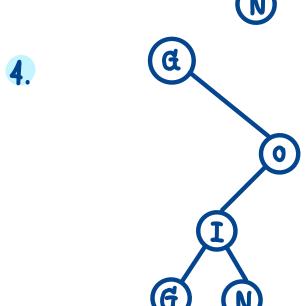


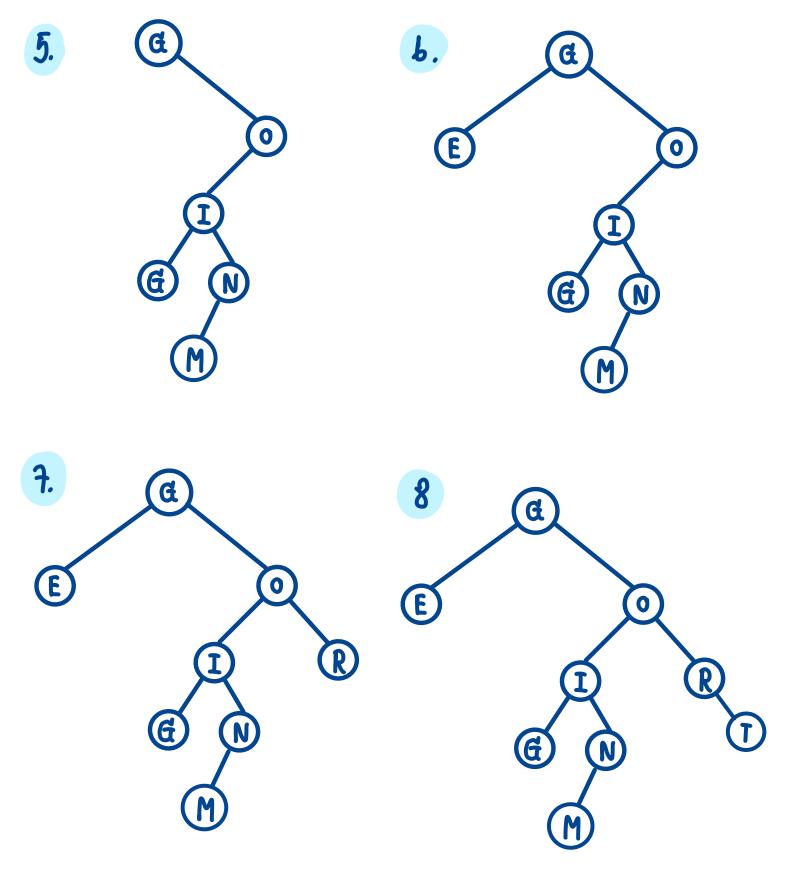


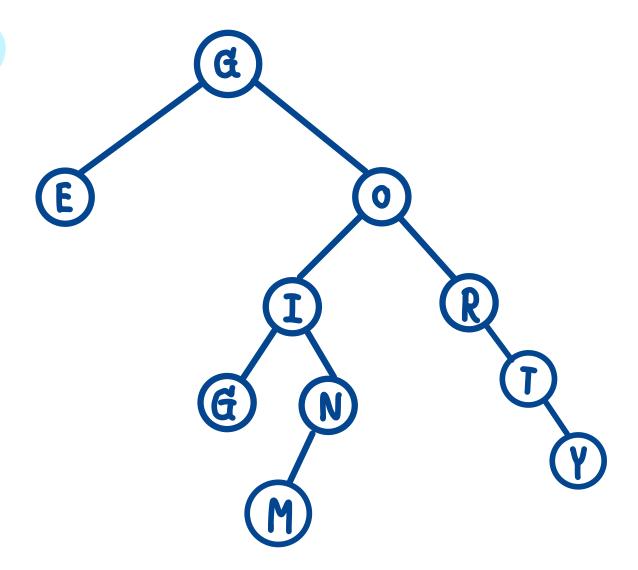
3. จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบ คำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว (ออกแบบบรรทัดเองเลยครับ)

```
BST tree2;
0.
      tree2.insert('G');
1.
      tree2.insert('0');
2.
      tree2.insert('I');
3.
4.
      tree2.insert('N');
      tree2.insert('G');
5.
6.
      tree2.insert('M');
7.
      tree2.insert('E');
      tree2.insert('R');
8.
      tree2.insert('T');
9.
      tree2.insert('Y');
10.
```

```
หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น ... GEOIENMRTY
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น ... EGMN IYTROG
```

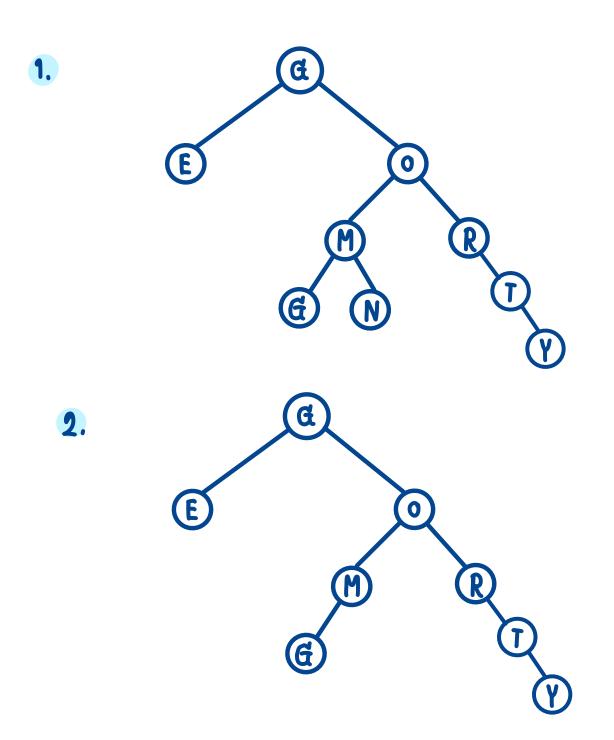




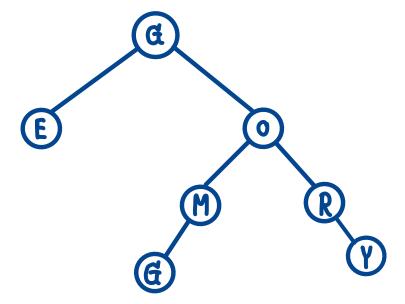


4. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ใน โปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

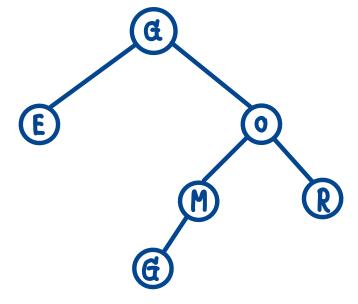
```
11. delete_node(&(tree2.root->right->left)); I
12. delete_node(&((tree2.root->right->left)->right)); N
13. delete_node(&((tree2.root->right->right)->right)); T
14. delete_node(&((tree2.root->right->right)->right)); Y
```







4

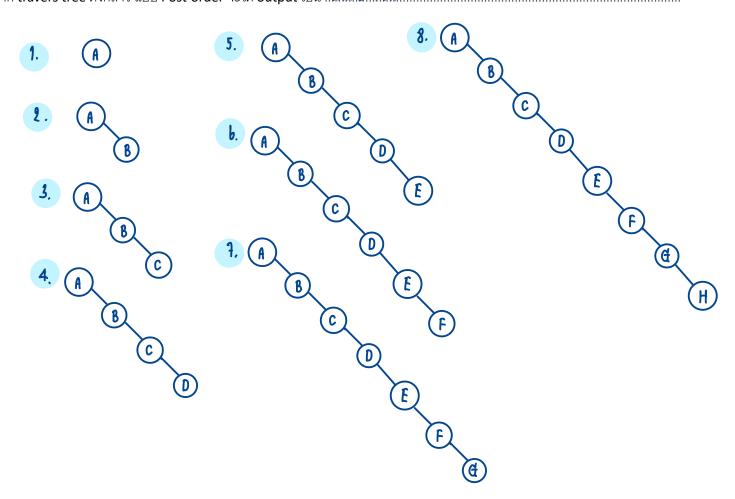


หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น ... **EEG MOR**หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น ... **EEG MOR**หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น ... **EEG MOR**

5. จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบ คำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว (ออกแบบบรรทัดเองเลยครับ)

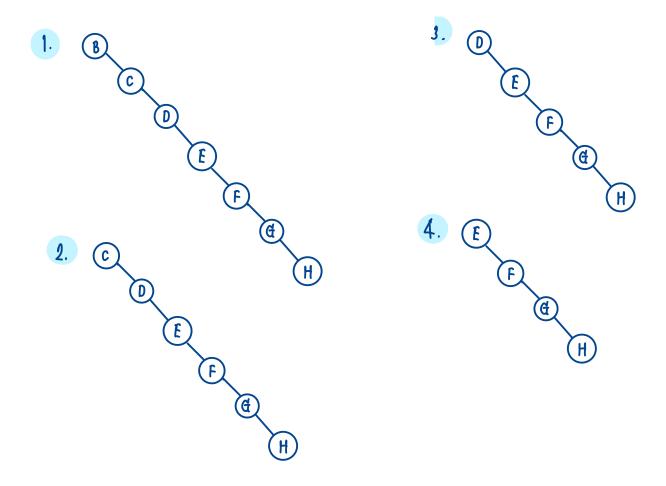
```
BST tree3;
1.
2.
      tree3.insert('A');
      tree3.insert('B');
3.
      tree3.insert('C');
4.
      tree3.insert('D');
5.
      tree3.insert('E');
6.
      tree3.insert('F');
7.
      tree3.insert('G');
8.
      tree3.insert('H');
9.
```

หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น ... ABCDEF&H
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น ... ABCDEF&H
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น ... HGFEDCBA



6. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ใน โปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

```
10. delete_node(&(tree3.root)); A
11. delete_node(&(tree3.root)); B
12. delete_node(&(tree3.root)); C
13. delete_node(&(tree3.root)); D
```



7.	BST ที่ balance กับ BST ที่ไม่ balance แบบใหนมีลำดับชั้นที่มากกว่ากัน หากจำนวนสมาชิกเท่ากัน เนื่องจากอะไร (ขอสั้นๆ) BST balance มีลำดับชั้นมากกว่า BST ไม่ balance เพราะไม่มีการเล่น node ไปส่วใก้ฝั่งหนึ่ง
8.	BST ที่ balance กับ BST ที่ไม่ balance หากต้องการ search แบบใหน ให้เวลาในการค้นหาน้อยกว่ากับ อย่างไร (ขอสั้นๆ) balance ดีกว่า เพาะมีการจักเรียงข้อมูล
9.	Tree ที่ balance กับ tree ที่ไม่ balance แบบใดโดยทั่วไปจะมีประสิทธิภาพดีกว่ากัน (ขอ1 คำ) balance
10.	. ดังนั้นการคิด algorithm และ data structure เราควรพยายามให้ tree อยู่ในรูปของ balance หรือ unbalance เนื่องจากอะไร (ขอยาวๆ) เพื่อให้ code กำงานได้ไวและมีระเบียบในการแก้ไขและทำงาน โกย balance จะทำงานเร็วกว่า unbalance