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');
```

H

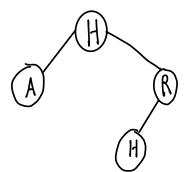
1.

2.

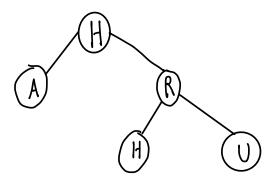
3.

A

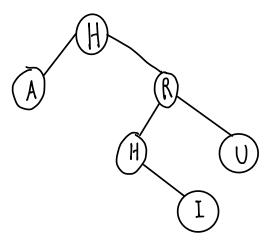
A R



5.



6.

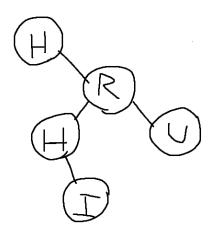


หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น	A	R	Н	Ī	U
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น	АНН	I R U			
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น	I	H U	R	H	

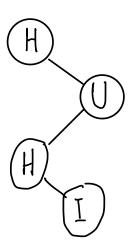
2. ต่อจากข้อ 1 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ใน โปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

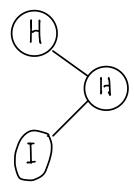
```
7.delete_node(&(tree.root->left));// A
8.delete_node(&(tree.root->right));
9.delete_node(&(tree.root->right));
```

7.



8.



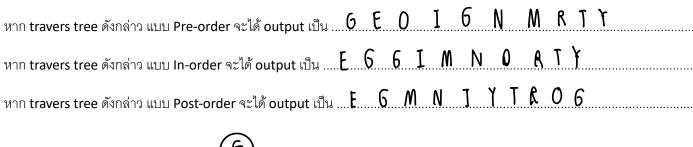


หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น HHI.

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

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

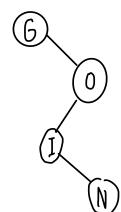
```
BST tree2;
0.
      tree2.insert('G');
1.
      tree2.insert('0');
2.
      tree2.insert('I');
3.
4.
      tree2.insert('N');
      tree2.insert('G');
5.
      tree2.insert('M');
7.
      tree2.insert('E');
      tree2.insert('R');
8.
      tree2.insert('T');
9.
      tree2.insert('Y');
10.
```

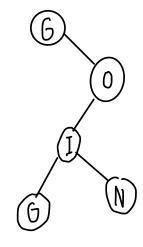


1)

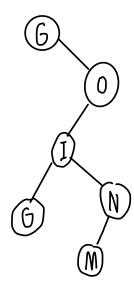


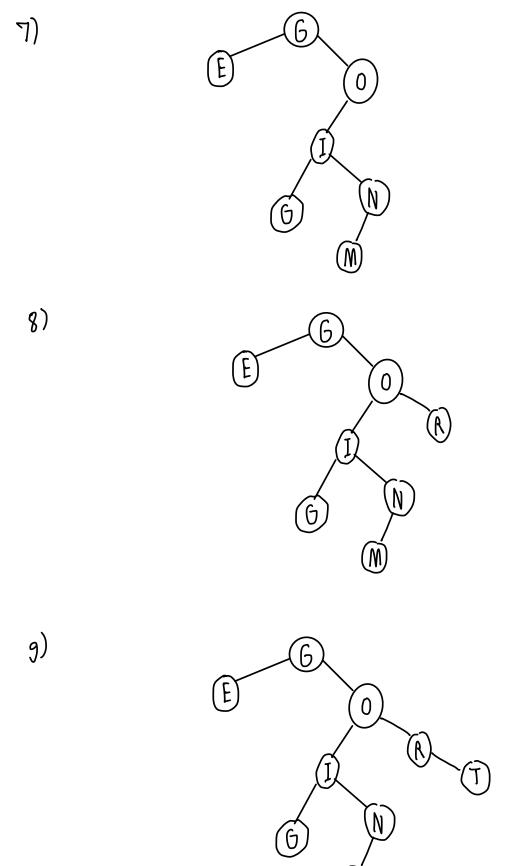


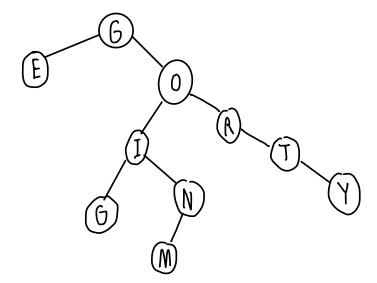




6)



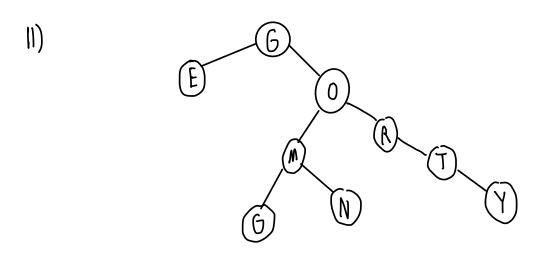


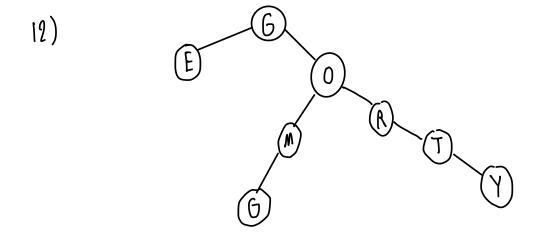


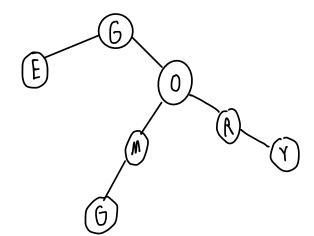
หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น 6 E 0 I 6 N M R T Y
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น E 6 6 I M N 0 A T Y
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น E 6 M N I Y T & O 6

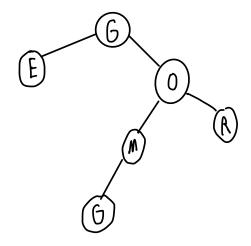
4. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ใน โปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

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









หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น 6 E 0 M 6 R
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น E 6 6 M 0 R
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น E 6 M R 0 6

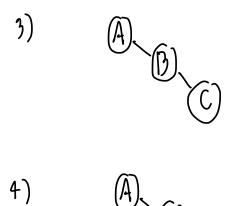
5. จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบ คำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว (ออกแบบบรรทัดเองเลยครับ)

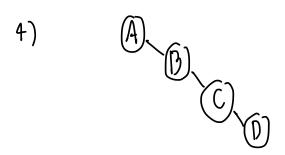
```
BST tree3;
1.
2.
      tree3.insert('A');
      tree3.insert('B');
3.
      tree3.insert('C');
4.
      tree3.insert('D');
5.
      tree3.insert('E');
      tree3.insert('F');
7.
      tree3.insert('G');
      tree3.insert('H');
9.
```

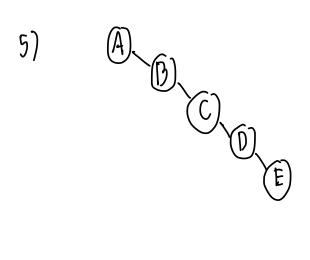
หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น A BC D E F 6 H
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น A BC D F F 6 H
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น H 6 F F D C BA

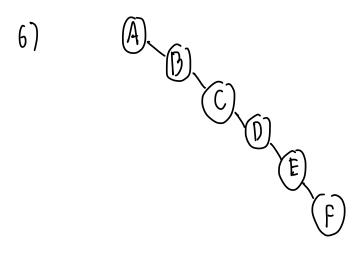








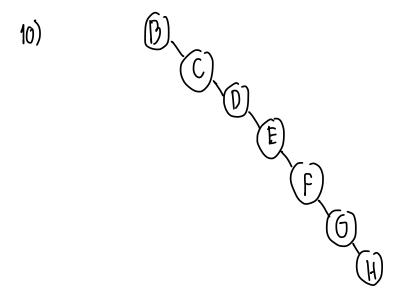


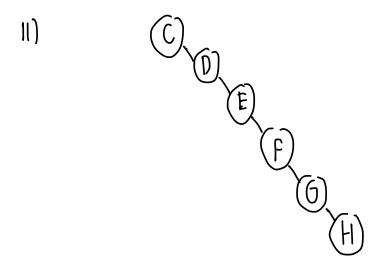


7) (A) (B) (C) (D) (E) (G)

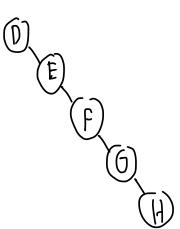
6. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ใน โปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

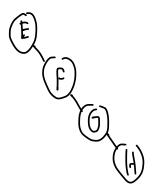
```
10. delete_node(&(tree3.root));
11. delete_node(&(tree3.root));
12. delete_node(&(tree3.root));
13. delete_node(&(tree3.root));
```











หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น E F G H
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น E F G H
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น H G F E

	BST ที่ balance กับ BST ที่ไม่ balance แบบใหนมีลำดับชั้นที่มากกว่ากัน หากจำนวนสมาชิกเท่ากัน เนื่องจากอะไร (ขอสั้นๆ) BST ที่ไม่ balance มีลำดับชั้นมากก่า เพเาะ แบบไม่ balance จะมีปัญวุลแต่ลงฝั่งไม่เท่ากัน โลยทำให้มีถ้าผนหึ่งยากว่าทำให้มีกำกับขั้นมากก่า
	BST ที่ balance กับ BST ที่ไม่ balance หากต้องการ search แบบใหน ให้เวลาในการค้นหาน้อยกว่ากัน อย่างไร (ขอสั้นๆ) BST ที่ balance ให้เวลาในการกันหาน่อยกว่าเพราะมีลำลับขั้นนับยุกว่า
9.	Tree ที่ balance กับ tree ที่ไม่ balance แบบใดโดยทั่วไปจะมีประสิทธิภาพดีกว่ากัน (ขอ1 คำ) B alancl
	ดังนั้นการคิด algorithm และ data structure เราควรพยายามให้ tree อยู่ในรูปของ balance หรือ unbalance เนื่องจากอะไร (ขอยาวๆ) เราการพยายามให้ tree อยู่ในรูปของ balance เนื่องจาก tree ในเปของ balance ทั้นจะที่ประทีทรัภพพากกว่าเพราะ สามาเกกันนาทั้งมนกโก้เร็วกว่า tree ในกูปของ unbalance เนื่องจาก tree ที่อยู่ในเปของ balance เมื่อเทียบกับ tree ที่อยู่ในเปของ unbalance นั้นจะมีจำนวน element นั่งยกว่าทากมีจำนาหกันก็กที่เก่ากัน