

Binary Search Tree

1. จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (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');
```

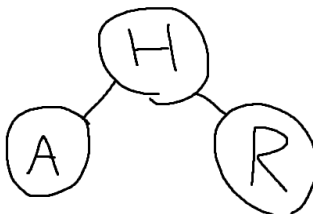
1.



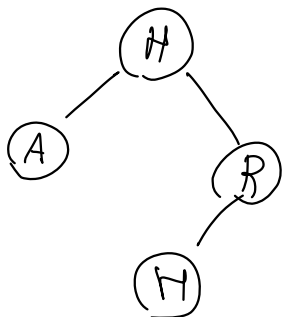
2.



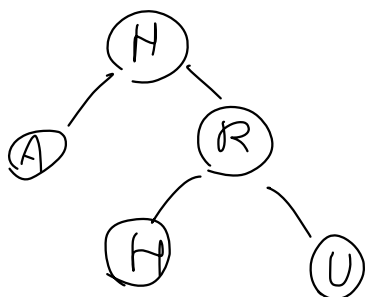
3.



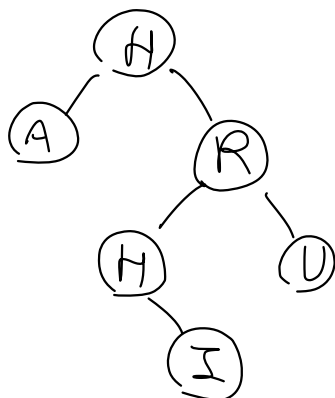
4.



5.



6.



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

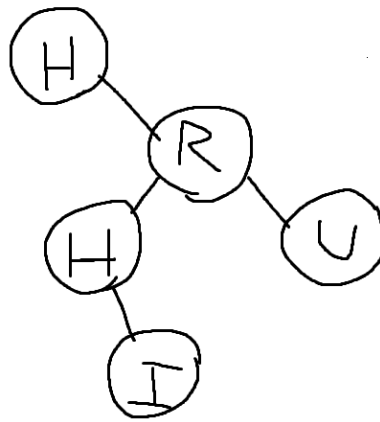
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น A H H I R U

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

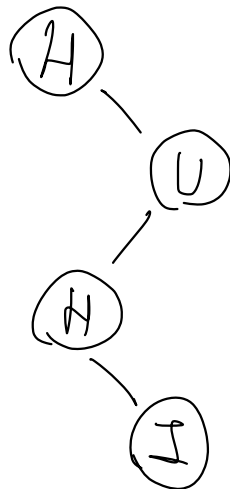
2. ต่อจากข้อ 1 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

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

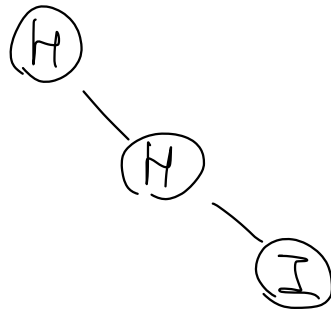
7.



8.



9.



หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น H H I

หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น H H I

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น I H H

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

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

หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็นGEOIGNMRTY.....

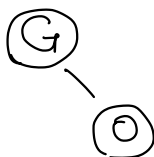
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็นEGGINORTY.....

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็นEGMNIYTROG.....

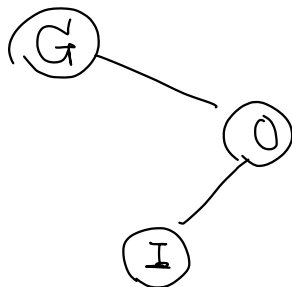
1



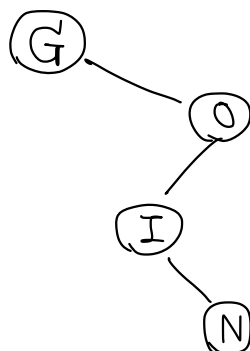
2



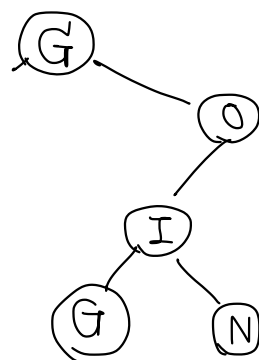
3



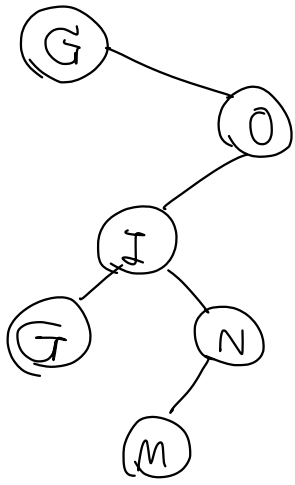
4



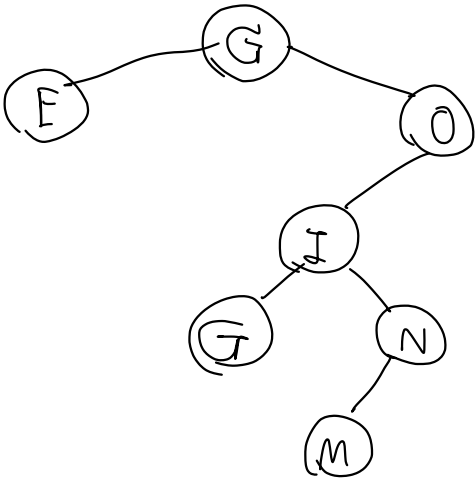
5



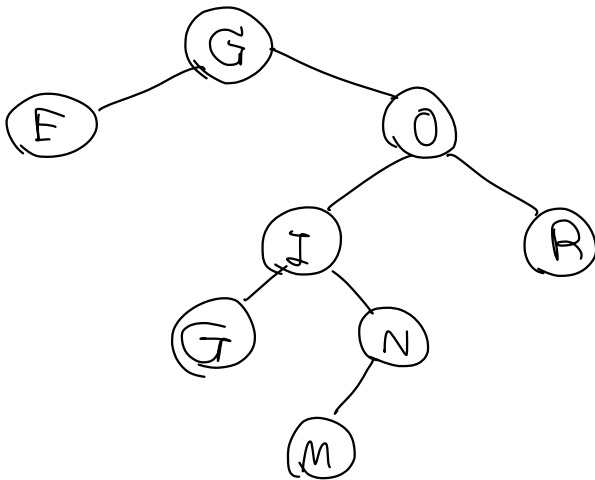
6



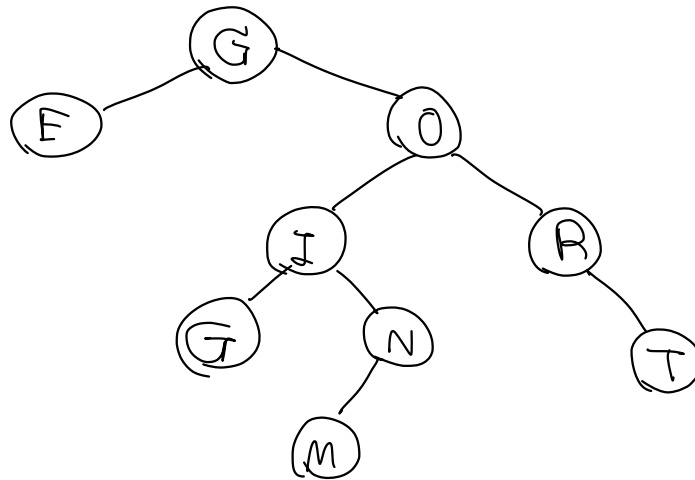
7



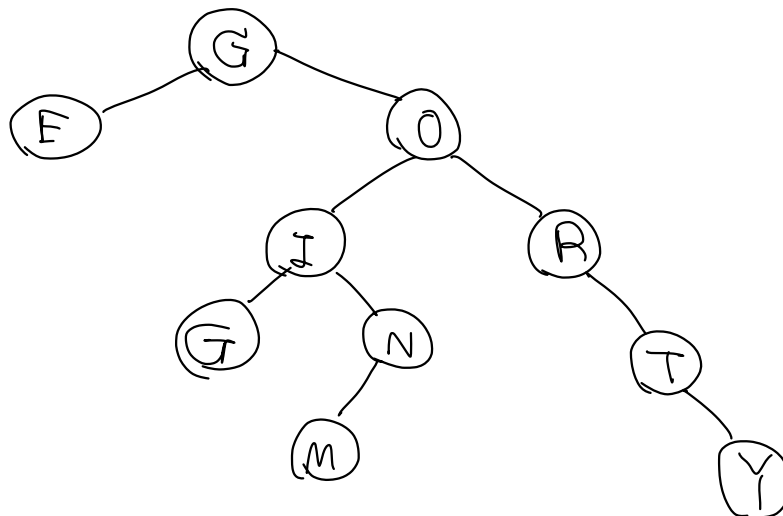
8



9



10



หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็นGEOIGNMRTY.....

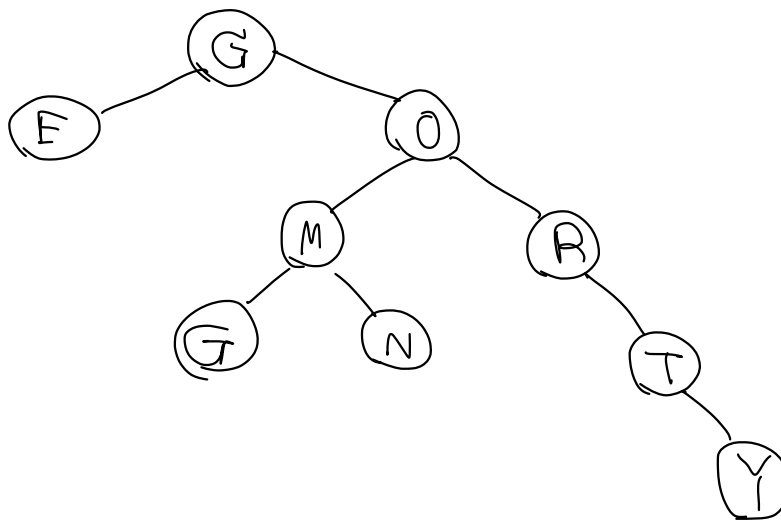
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็นEGGIMNORTY.....

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็นEGMNIYTROG.....

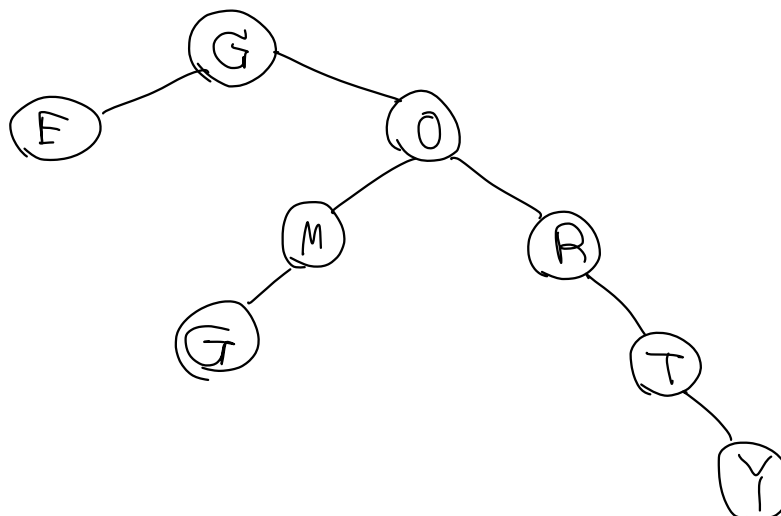
4. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

```
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));
```

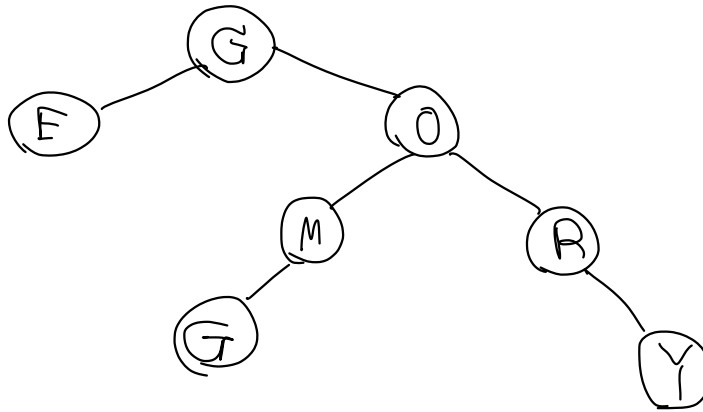
11



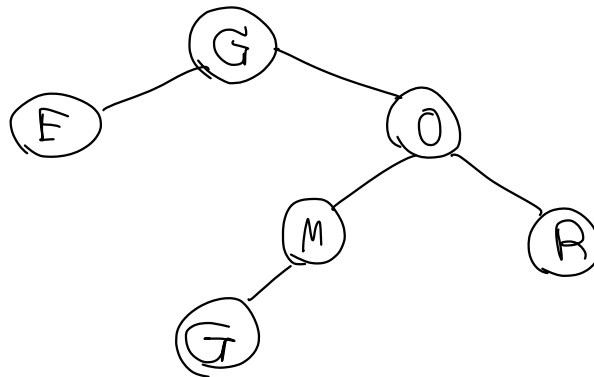
12



13



14



หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น GEOMGR

หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น EGGMOR

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น EGMR OG

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

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

หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็นA B C D E F G H.....

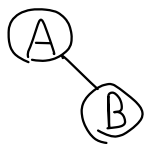
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็นA B C D E F G H.....

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็นH G F E D C B A.....

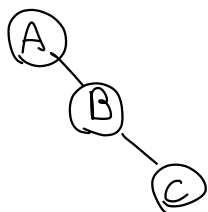
2



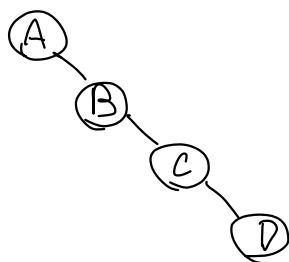
3



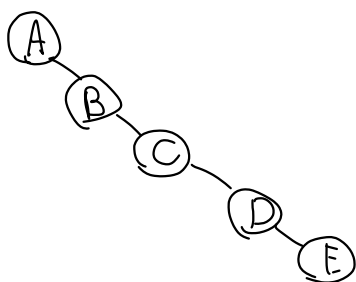
4



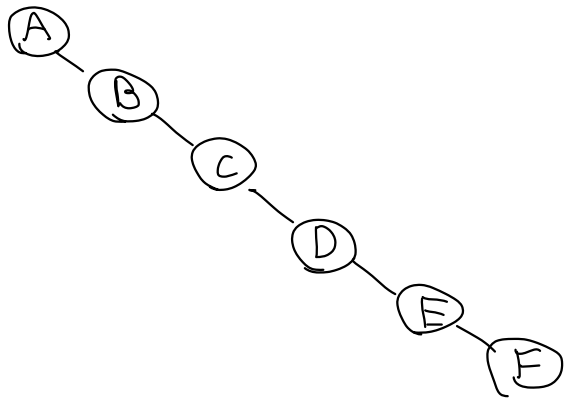
5



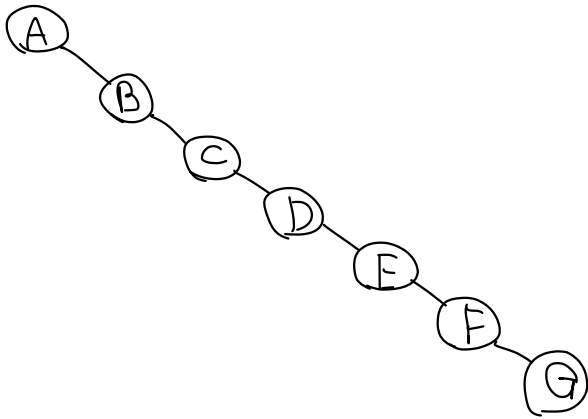
6



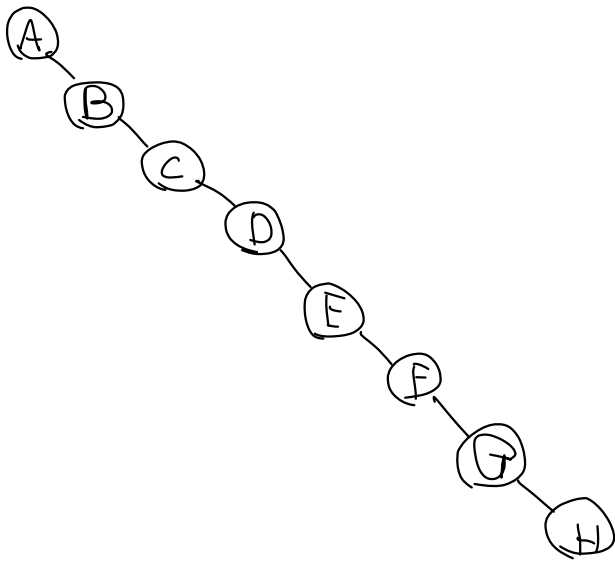
7



8



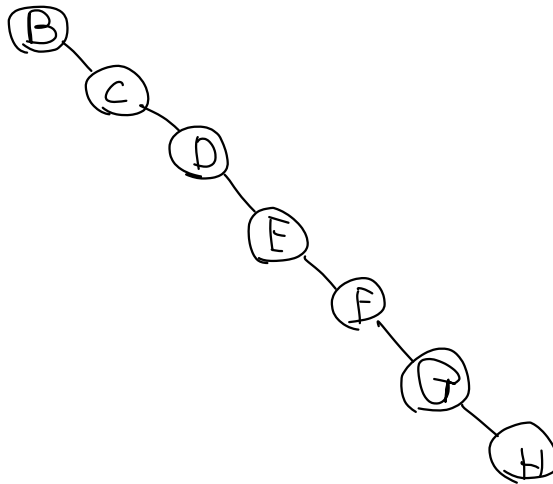
9



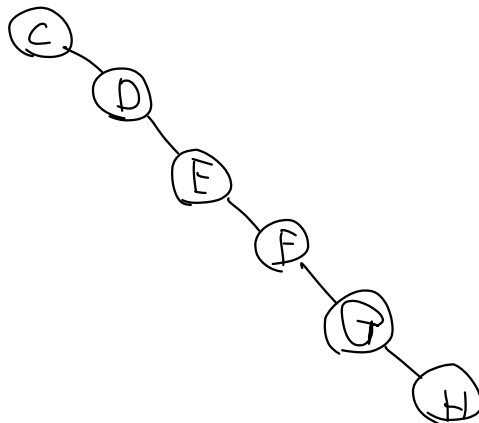
6. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

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

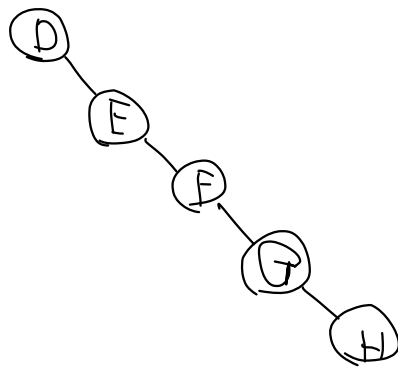
10



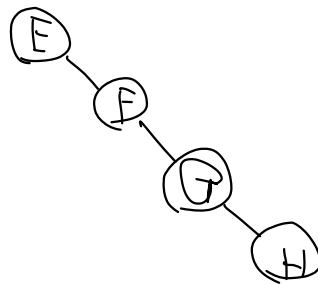
11



12



13



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

7. BST ที่ balance กับ BST ที่ไม่ balance แบบไหนมีลำดับชั้นที่มากกว่ากัน หากจำนวนสมาชิกเท่ากัน เนื่องจากอะไร (ขอสั้นๆ)

ไม่ Balance มีลำดับชั้นมากกว่า เนื่องจาก เมื่อ Balance จะทำให้
สมดุล เรียงกัน จนกว่า ชั้นนั้นจะเต็ม แต่เมื่อไม่ Balance จะเรียง
แบบใดก็ได้ ธรรมชาติของอยู่ใน BST rule

8. BST ที่ balance กับ BST ที่ไม่ balance หากต้องการ search แบบไหน ให้ความเร็วในการค้นหาน้อยกว่ากัน อย่างไร (ขอสั้นๆ)

เมื่อ Balance เพราะ เมื่อ Balance สมดุล จะทำให้ มี ลำดับชั้นน้อยลง
เนื่องจาก การ search จะ search ไล่ชั้น ๆ ไป (ห้อยไปซ้าย , ขวาไปขวา)

9. Tree ที่ balance กับ tree ที่ไม่ balance แบบใดโดยทั่วไปจะมีประสิทธิภาพดีกว่ากัน (ขอ 1 คำ)

Balance

10. ดังนั้นการคิด algorithm และ data structure เราควรพยายามให้ tree อยู่ในรูปของ balance หรือ unbalance เนื่องจากอะไร (ขอยาวๆ)

เมื่อ Balance เนื่องจาก เมื่อ สร้าง tree เมื่อเมื่อ Balance จะทำให้
ลำดับชั้นของ tree ลดลง เมื่อ เราทำการ search โดยใช้ BST rule
จะคิดเพียง 1-2 ชั้น แต่ถ้าไม่ Balance จะมีความลึกได้ ทำให้ ไล่ลงมาจนกว่า
เจอค่าที่ต้องการ เมื่อ ไม่ Balance จะ search ไล่เรื่อย ๆ ทำให้
ไล่ลงมาจนกว่า เพราะ ถ้า ชั้น มีค่ามากกว่า 100 ก็จะต้อง search
จนเจอค่าที่ต้องการ