ดลพพร พันธุ์ 65070339

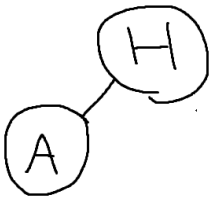
Binary Search Tree

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

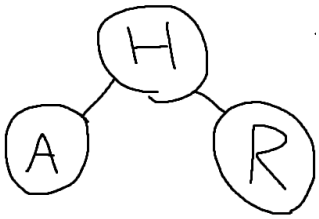
```
0.    BST tree;
1.    tree.insert('H');
2.    tree.insert('A');
3.    tree.insert('R');
4.    tree.insert('H');
5.    tree.insert('U');
6.    tree.insert('I');
```

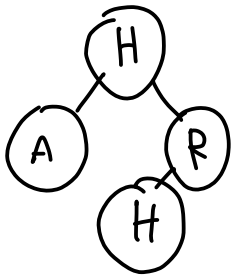
1.

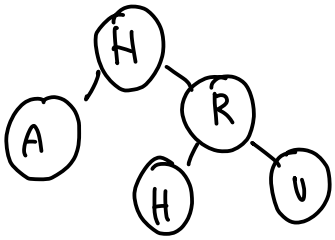
2.

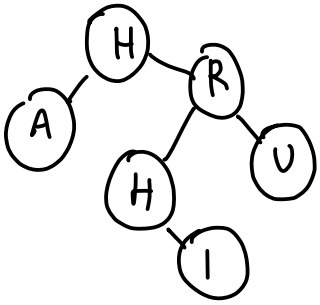


3.



4. 

5. 

6. 

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

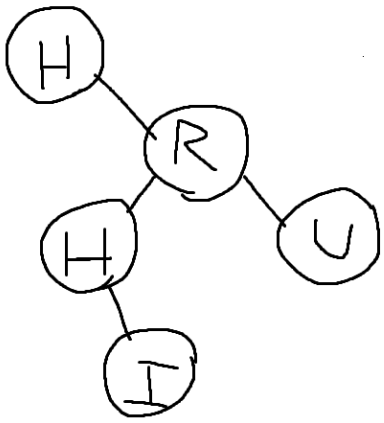
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น ..... A H H I R U .....

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

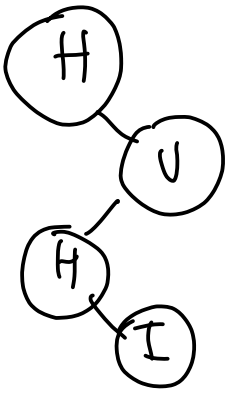
2. ต่อจากข้อ 1 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

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

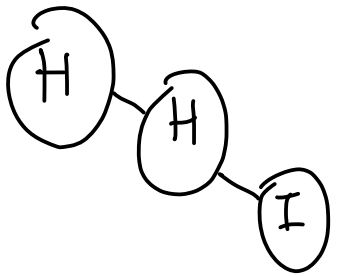
7.



8.



9.



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

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

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

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 เป็น GEOIGNMRTY

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

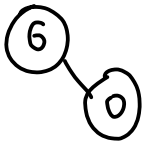
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น EGMNIYTROG

GOINGMERTY

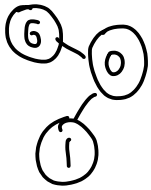
ABCDEFGHIJKLMNOP
QRSTUVWXYZ

1

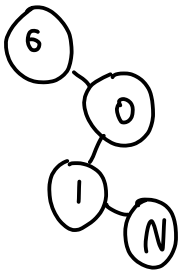
2



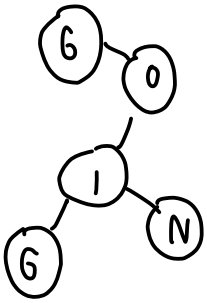
3



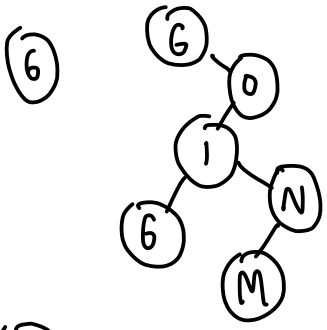
4



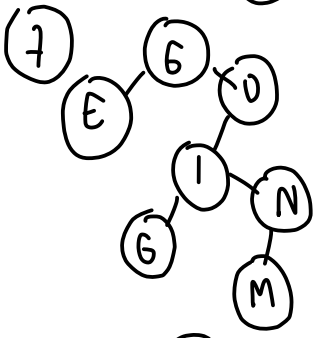
5



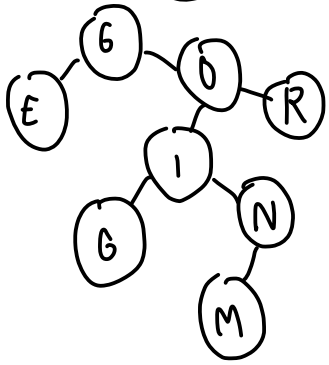
6



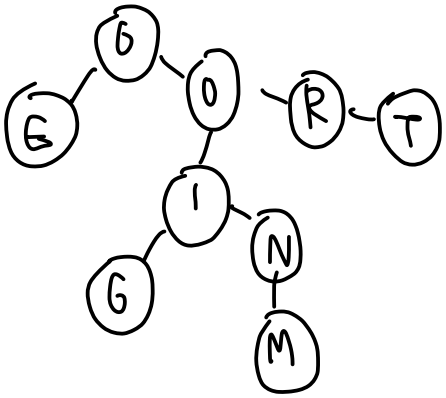
7



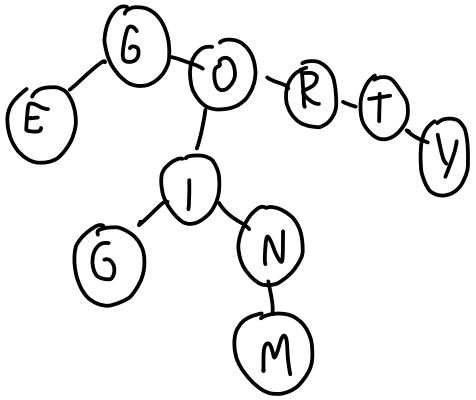
8



9



10



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

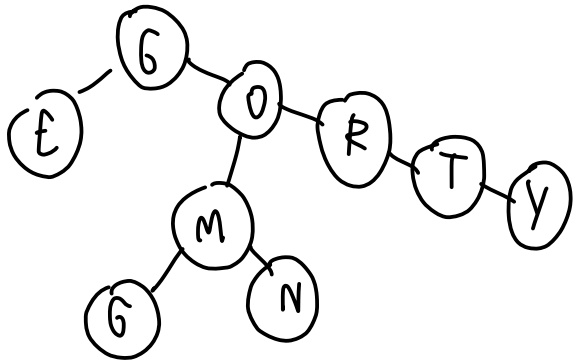
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น .....

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

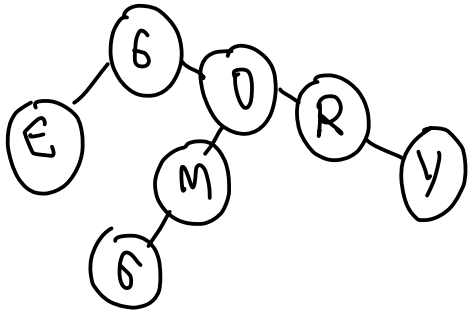
4. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

```
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));
```

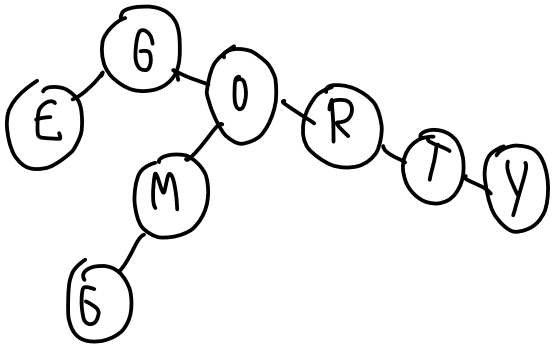
11



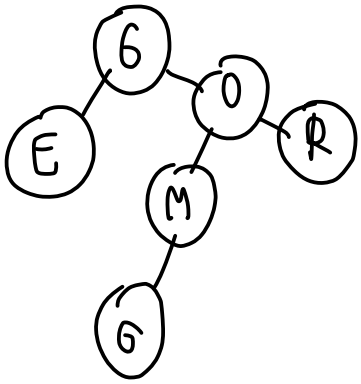
13

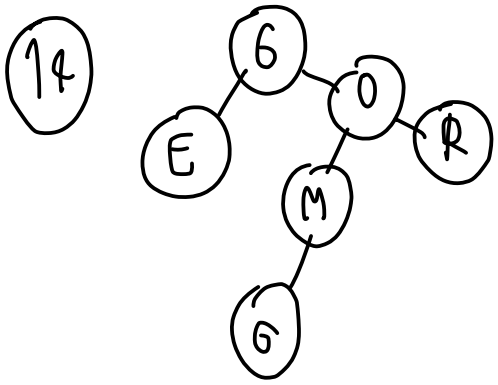


12



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 EGMROG

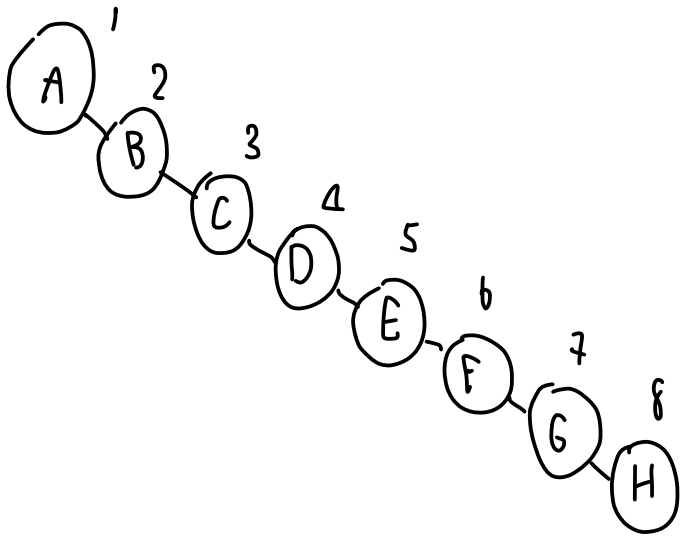
5. จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว (ออกแบบบรรทัดเองเลยครับ)

```
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 เป็น ABCDEFGH

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

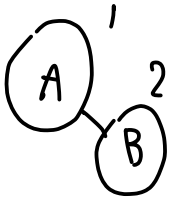
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น HGFEDCBA



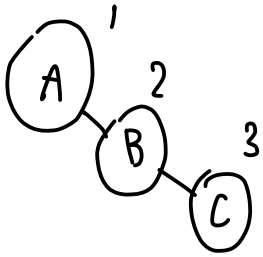
1


A
1

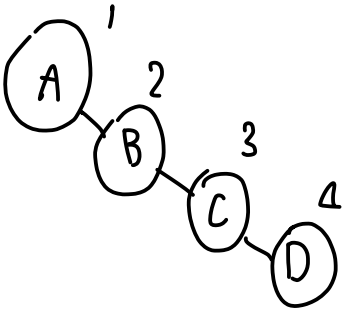
2


A
1
B
2

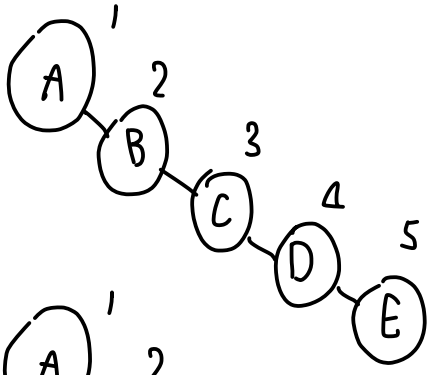
3


A
1
B
2
C
3

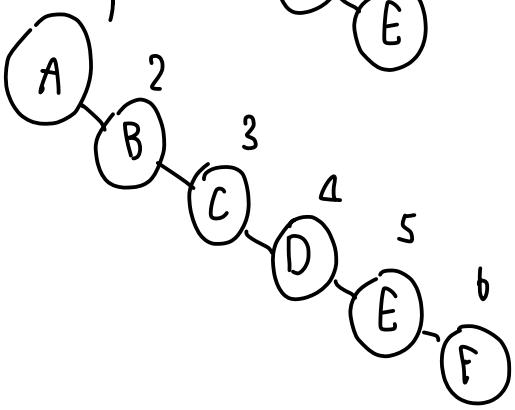
4


A
1
B
2
C
3
D
4

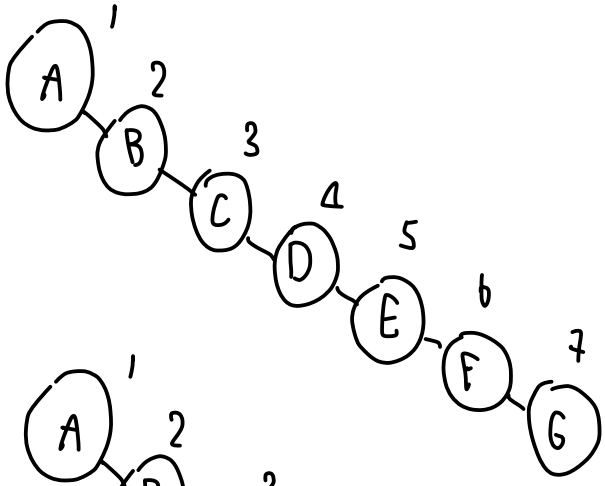
5


A
1
B
2
C
3
D
4
E
5

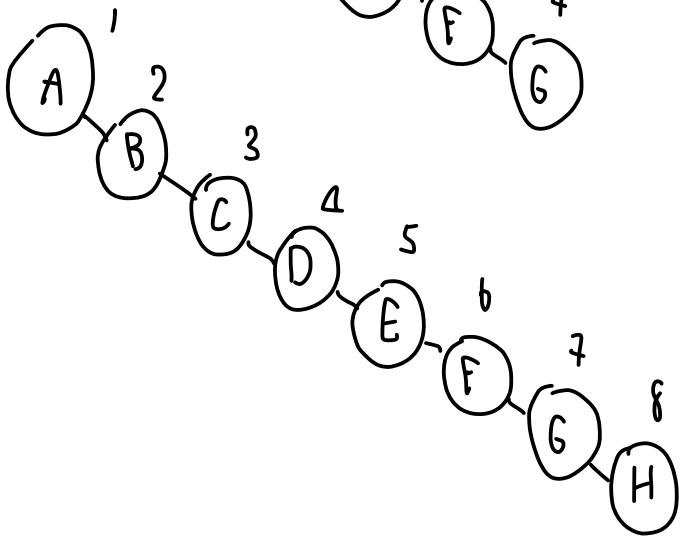
6


A
1
B
2
C
3
D
4
E
5
F
6

7

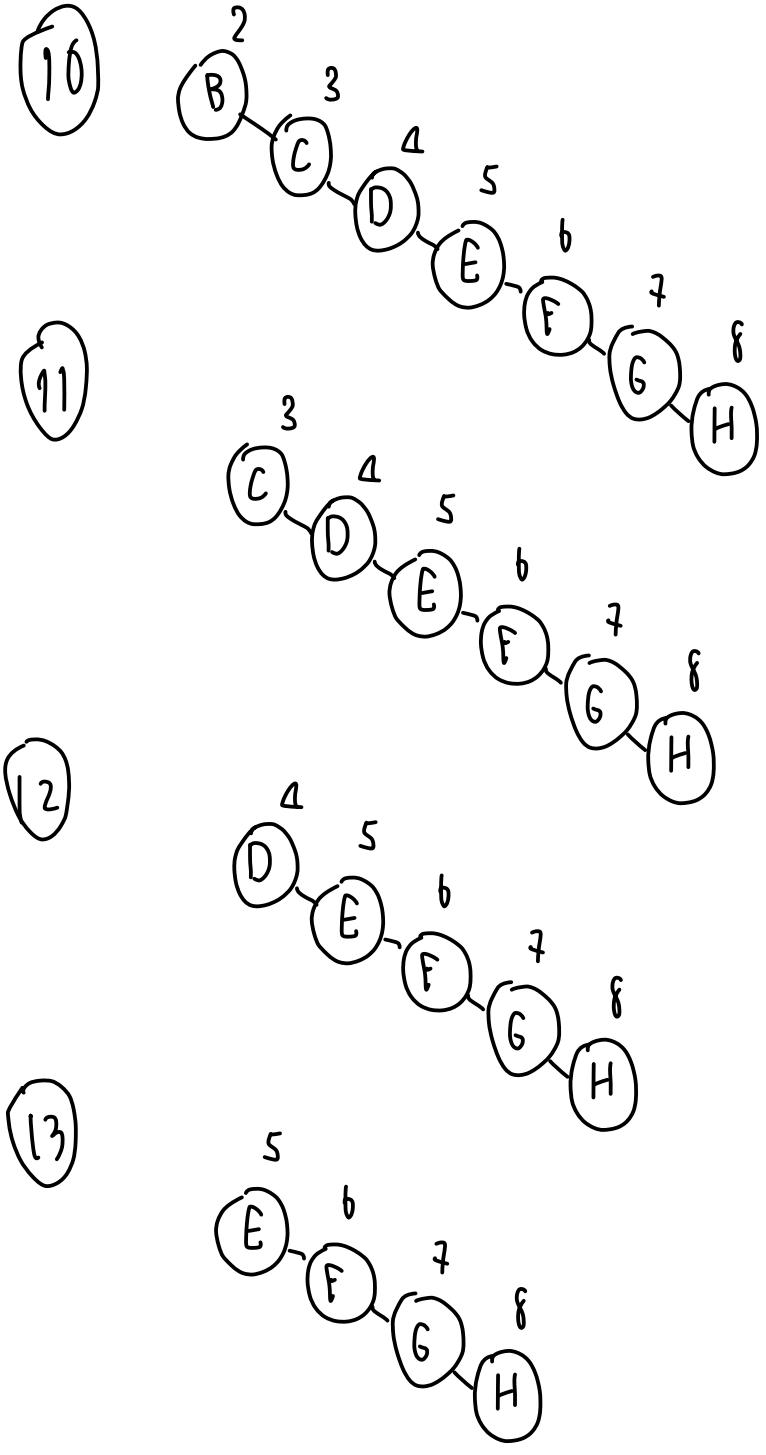

A
1
B
2
C
3
D
4
E
5
F
6
G
7

8


A
1
B
2
C
3
D
4
E
5
F
6
G
7
H
8

6. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ 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));
```



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

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

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น HGFE

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

ไม่ balance มีลำดับชั้นมากกว่าเพราะ ในจำนวนสมาชิกที่เท่ากัน balance จะมีลำดับชั้นด้านซ้าย=ด้านขวา แต่ unbalance มันจะไปกองอยู่ด้านใดด้านหนึ่ง ทำให้ลำดับชั้นมากกว่า

8. BST ที่ balance กับ BST ที่ไม่ balance หากต้องการ search แบบใหน ให้เวลาในการค้นหาน้อยกว่ากัน อย่างไร (ขอสั้นๆ)

balance BST ใช้เวลาในการค้นหาน้อยกว่าเพราะเมื่อแยกค่ามากกว่า น้อยกว่า ไงล่ะ มีลำดับชั้นน้อยกว่า จะทำให้ใช้เวลาน้อยกว่า

9. Tree ที่ balance กับ tree ที่ไม่ balance แบบใดโดยทั่วไปจะมีประสิทธิภาพดีกว่ากัน (ขอ1 คำ)

balance tree

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

ทำให้ balance เพราะเมื่อเรา balance ข้อมูลจะถูกจัดไว้เป็นระเบียบมากกว่า ทำให้เวลาหาข้อมูลที่มีค่ามาก ไม่จำเป็นต้องเข้าถึงข้อมูลทุกตัว ก็สามารถหาข้อมูลค่านั้นๆเจอได้ หากเป็นแบบ unbalance อาจต้องเข้าถึงข้อมูลทุกตัวก่อน จะเจอตัวที่ต้องการค้นหา