Binary Search Tree

นายเศรษฐกานต์ พีรเศวตพงศ์ 65011068

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.

A picture containing sketch, drawing, clipart, line art

Description automatically generated

2.

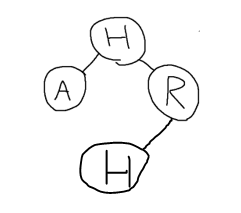
A picture containing sketch, drawing, line art, clipart

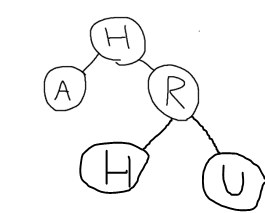
Description automatically generated

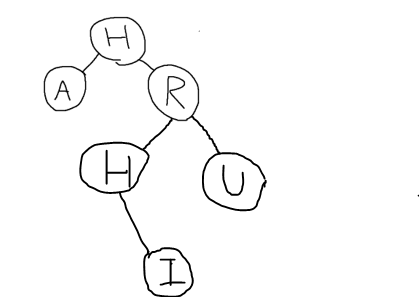
3.

A picture containing sketch, line art, drawing, clipart

Description automatically generated

4.

5.

6.

H A R H I U

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

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

A I H U R H

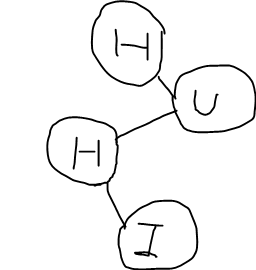
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น ................................................................................................................

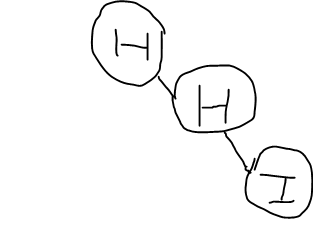
1. ต่อจากข้อ 1 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว
2. delete\_node(&(tree.root->left));// A
3. delete\_node(&(tree.root->right));
4. delete\_node(&(tree.root->right));

7.

A picture containing sketch, drawing, line art, clipart

Description automatically generated

8.

9.

H H I

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

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

I H H

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

1. 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');

G E O I G N M R T Y

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

E G G I M N O R T Y

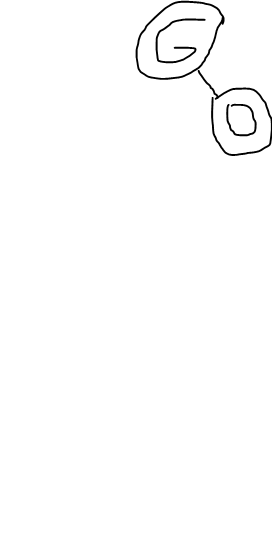
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น .............................................................................................................

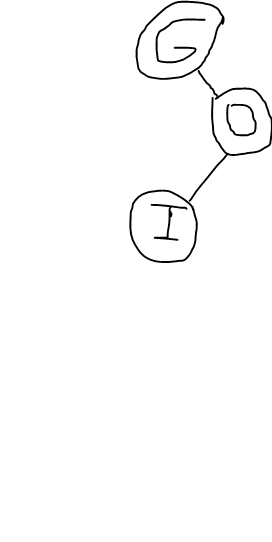
E G M N I Y T R O G

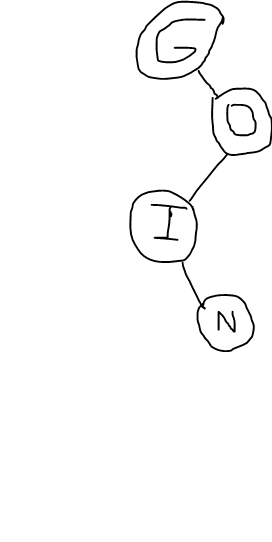
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น .........................................................................................................

1.

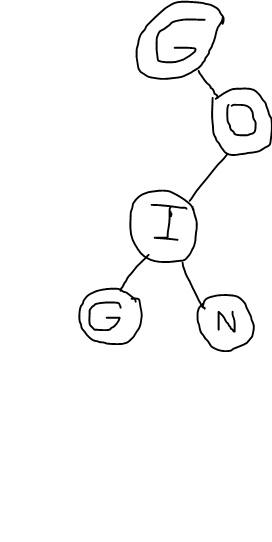
2.

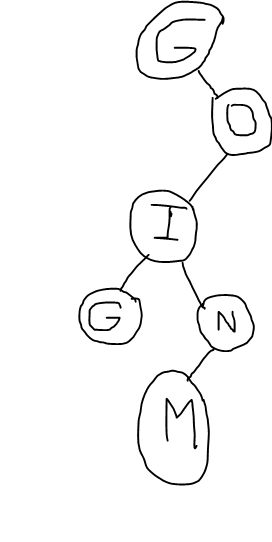


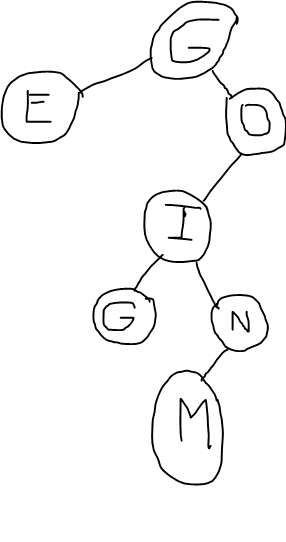
3.

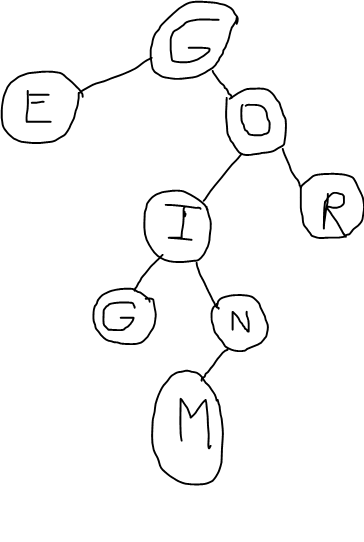


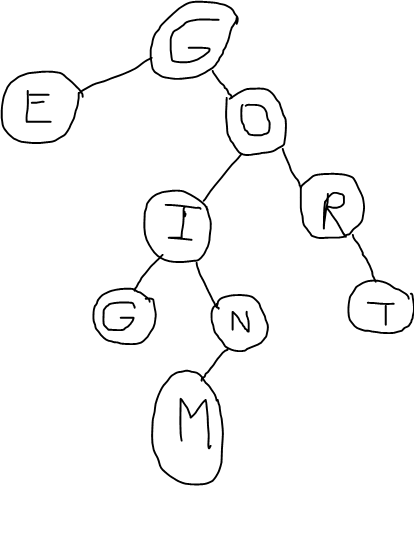
4.

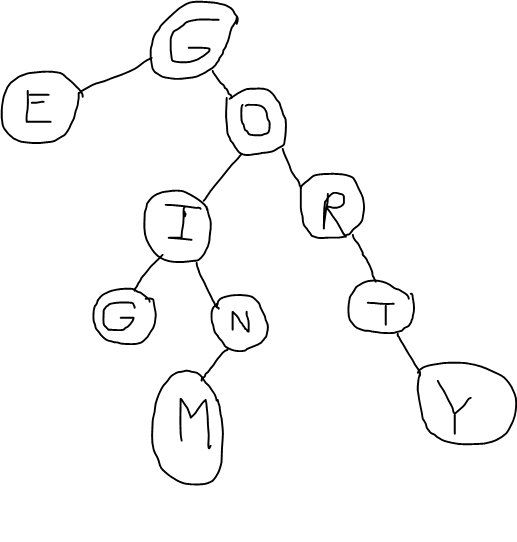
5.

6.

7.

8.

9.



10.

G E O I G N M R T Y

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

E G G I M N O R T Y

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

E G M N I Y T R O G

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

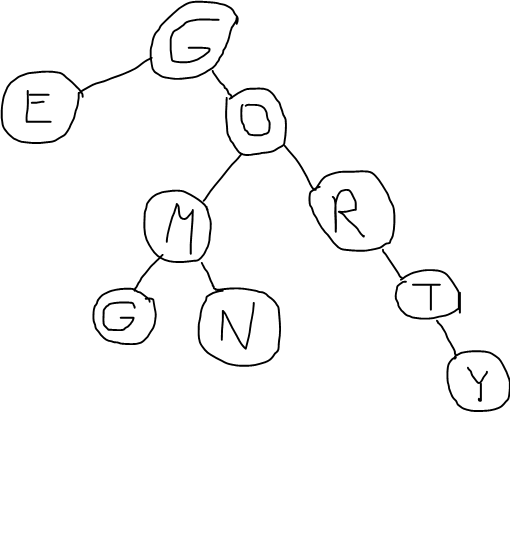
1. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

11.    delete\_node(&(tree2.root->right->left));

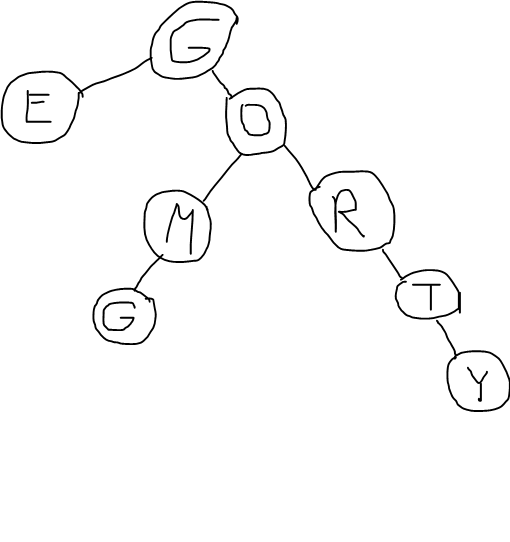
12.    delete\_node(&((tree2.root->right->left)->right));

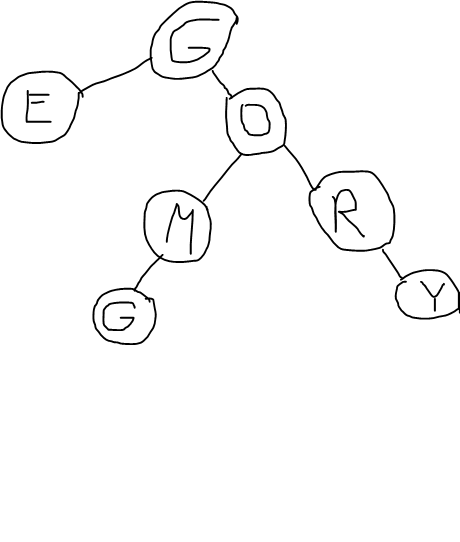
13.    delete\_node(&((tree2.root->right->right)->right));

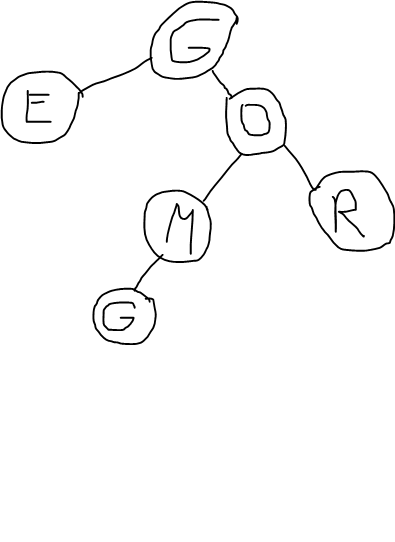
14.    delete\_node(&((tree2.root->right->right)->right));



1.

2.

3.

4.

G E O M G R

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

E G G M O R

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

E G M R O G

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

1. 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');

A B C D E F G H

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

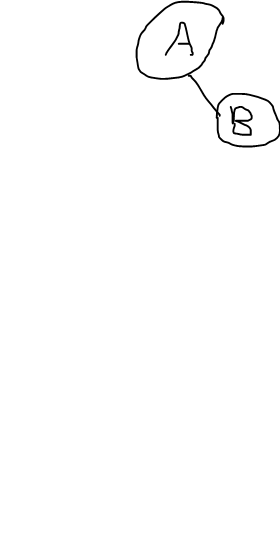
A B C D E F G H

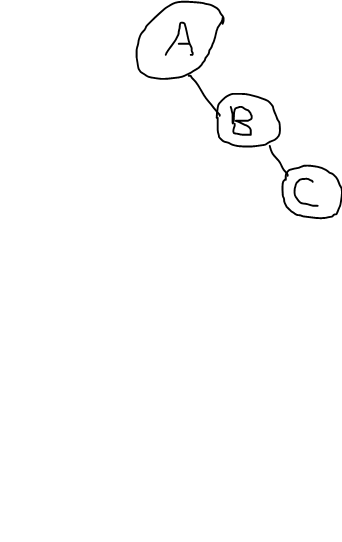
หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น .............................................................................................................

H G F E D C B A

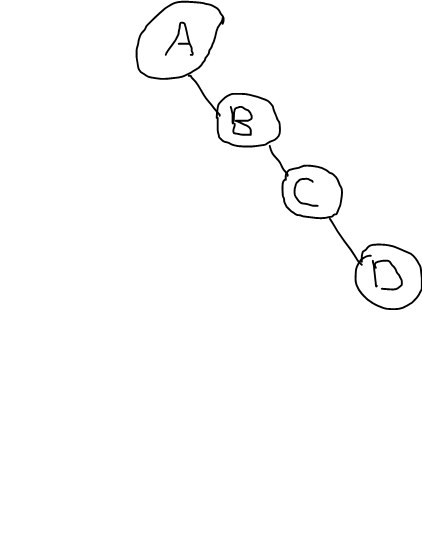
หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น ............................................ .............................................................

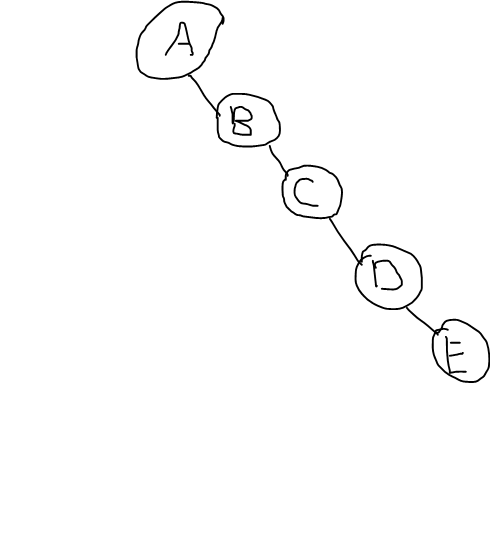
1.

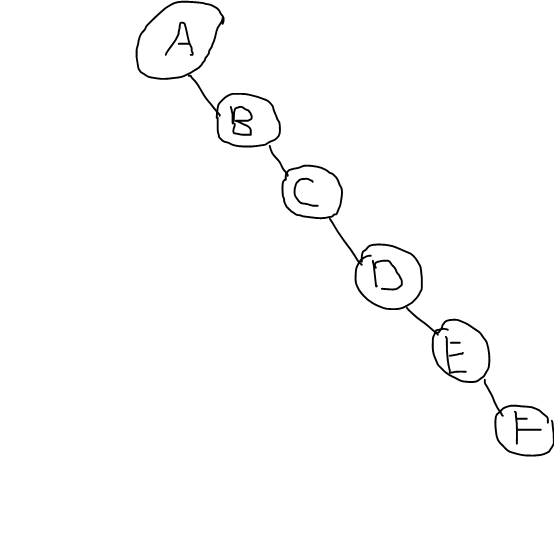
2.



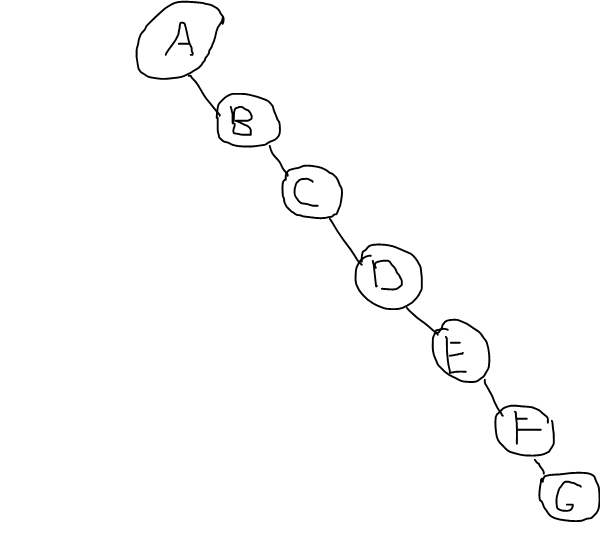
3.

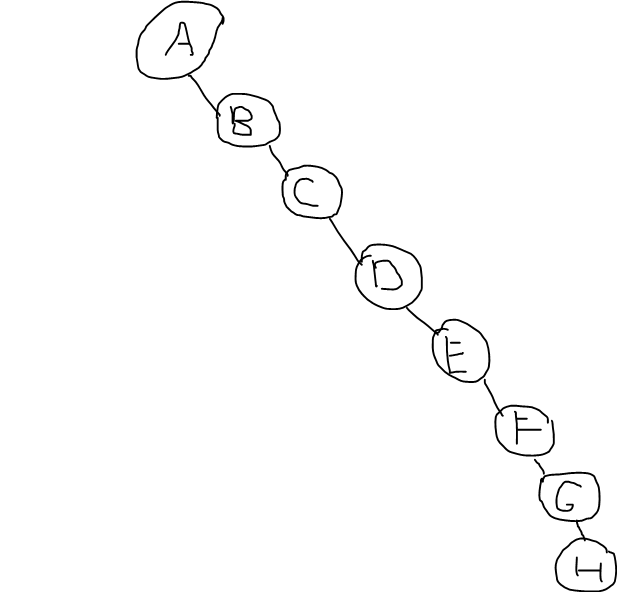
4.

5.



6.

7.

8.

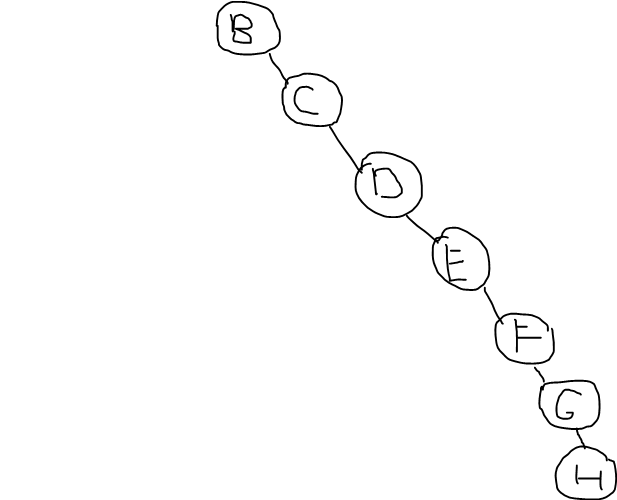
1. ต่อจากข้อ 3 หากใช้ code ดังต่อไปนี้ จงเขียนแผนภาพของการทำงานของ Binary search tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

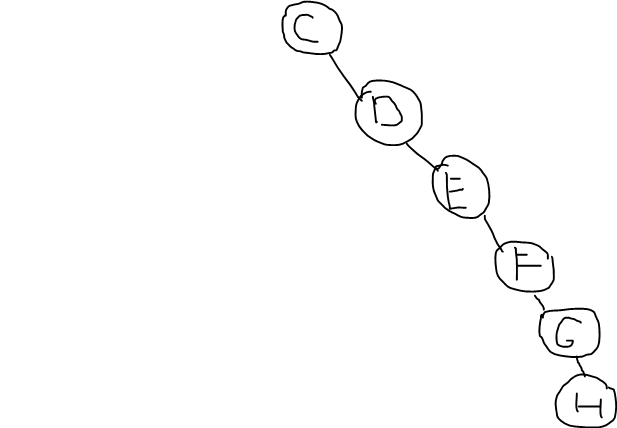
10.    delete\_node(&(tree3.root));

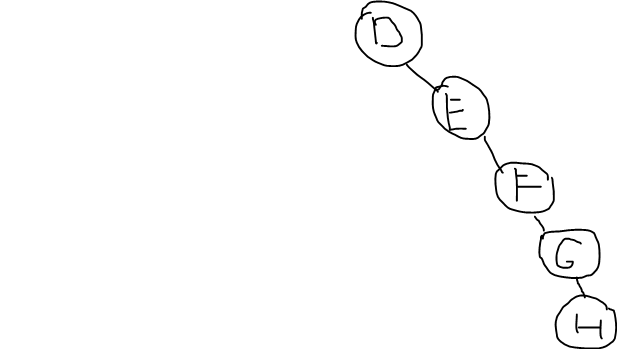
11.    delete\_node(&(tree3.root));

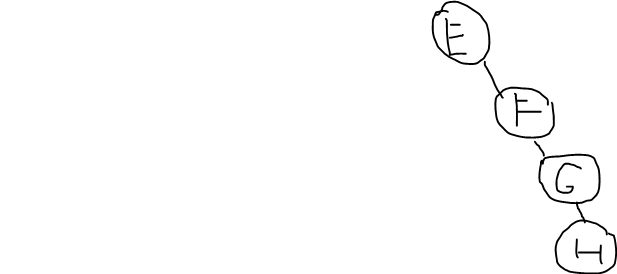
12.    delete\_node(&(tree3.root));

13.    delete\_node(&(tree3.root));

1.

2.

3.



4.

E F G H

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

E F G H

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

H G F E

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

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

BST แบบไม่ balance

เพราะเมื่อจำนวนสมากชิกเท่ากันการสร้าง BST ที่

ไม่ balance เกิดจากการกำหนดเงื่อนไขที่ไม่แน่

แน่นอนทำให้แบบที่ไม่ balance มี node เกิน

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

ชั้นที่น้อยกว่าเลยทำให้เวลาน้อยลงตามไปด้วย

BST แบบ balance ใช้เวลาน้อยกว่าเพราะมีลำดับ

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

balance

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

ควรจะใช้ tree ที่อยู่ในรูปของ balance เพราะการ

...................................................................................................................................................................................................................................................................................................................... ........................................................................................................................................................... ...........................................................................................................................................................

ในการ searchจะเร็วขึ้นเนื่องจากลำดับชั้นนั้นน้อย

ลำดับใช้ได้มากที่สุดจน balance ได้และความเร็ว

กำหนดในการสร้าง tree ทำออกมาได้ดีและลด

ทำให้ balance จะเป็นการแสดงให้เห็นว่าเงื่อนไขที่

...................................................................................................................................................................................................................................................................................................................... ........................................................................................................................................................... ...........................................................................................................................................................

สามารถดูและแยกแยะออกมาได้ง่าย

ตามลำดับชั้นของ tree แบบ balance ทำให้

...........................................................................................................................................................