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.

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.

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

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.

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

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

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');

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

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

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));

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

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');

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

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));

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

1. BST ที่ balance กับ BST ที่ไม่ balance แบบใหนมีลำดับชั้นที่มากกว่ากัน หากจำนวนสมาชิกเท่ากัน เนื่องจากอะไร (ขอสั้นๆ) ......................................................................................................................................................................................................................................................................................................................
2. BST ที่ balance กับ BST ที่ไม่ balance หากต้องการ search แบบใหน ให้เวลาในการค้นหาน้อยกว่ากัน อย่างไร (ขอสั้นๆ)  
   ......................................................................................................................................................................................................................................................................................................................
3. Tree ที่ balance กับ tree ที่ไม่ balance แบบใดโดยทั่วไปจะมีประสิทธิภาพดีกว่ากัน (ขอ1 คำ)  
   ...........................................................................................................................................................
4. ดังนั้นการคิด algorithm และ data structure เราควรพยายามให้ tree อยู่ในรูปของ balance หรือ unbalance เนื่องจากอะไร (ขอยาวๆ)

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

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

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