

Binary Search Tree: Traversal

ให้เขียนโปรแกรมรับเลขจำนวนเต็ม ไม่ซ้ำกัน จำนวน N ค่าเพื่อสร้าง Binary Search Tree และรับจำนวน operation และหมายเลข operation (OP_k) ของ Traversal แบบต่างๆ เพื่อแสดงค่าใน Binary Search Tree โดยที่

OP_k	การทำงาน (โดยตัวเลขแต่ละตัวคั่นด้วยช่องว่าง (space))
1	แสดงผลของการ Traverse ใน Tree แบบ preorder
2	แสดงผลของการ Traverse ใน Tree แบบ inorder
3	แสดงผลของการ Traverse ใน Tree แบบ postorder
4	แสดงผลของการ Traverse ใน Tree แบบ Breath first search แยกบรรทัดละ level โดยแต่ละบรรทัดจะแสดงผลเลขกำกับแต่ละ level ($0 \dots H-1$) ตามด้วยค่าข้อมูลใน level นั้น เมื่อ H เป็นจำนวน level ในทรี
5	แสดงค่าข้อมูลที่เป็น Leaf nodes เรียงจากซ้ายไปขวา

ข้อมูลเข้า มี $N+3$ บรรทัด

บรรทัดแรกจะเป็นเลขจำนวนเต็ม 1 ตัวแทนจำนวนโหนด N ใน Tree ($1 \leq N \leq 1000$)

บรรทัดที่ 2 ถึง $N+1$ จะเป็นเลขจำนวนเต็ม X_i แทนค่าของข้อมูลที่ต้องการเก็บลงใน Binary Search Tree โดยที่ $1 \leq X_i \leq 30000$ และ $i=1 \dots N$

บรรทัดที่ $N+2$ เป็นเลขจำนวนเต็ม 1 ตัวแทนจำนวน operation ที่ต้องการทำ ($1 \leq M \leq 5$)

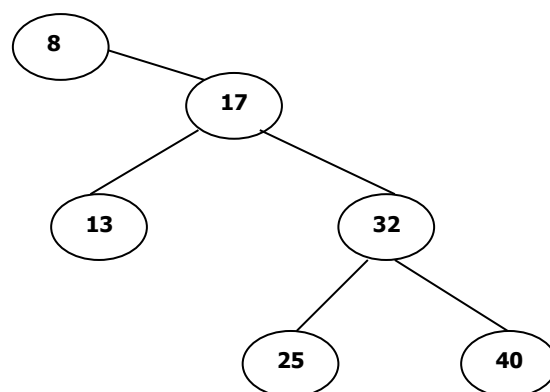
บรรทัดที่ $N+3$ เป็นเลขจำนวนเต็ม OP_k แทนหมายเลข operation ของ Traversal แบบต่างๆ ใน Binary Search Tree โดยที่ $1 \leq OP_k \leq 5$ และ $k = 1 \dots M$

ข้อมูลออก

มี M หรือ $M+H$ (กรณีทำ operation หมายเลข 4 เมื่อ H เป็นจำนวน level ในทรี) ซึ่งเป็นการแสดงผลการทำงานตามการทำงานของหมายเลข operation ที่รับเข้า (OP_k) ดังระบุในตารางข้างต้น โดยตัวเลขแต่ละตัวคั่นด้วยช่องว่าง (space)

ตัวอย่าง 1

ข้อมูลเข้า	ข้อมูลออก
6	8 17 13 32 25 40
8	8 13 17 25 32 40
17	13 25 40 32 17 8
32	
25	
13	
40	
1	
1 2 3	



ตัวอย่าง 2-3

ข้อมูลเข้า	ข้อมูลออก	ข้อมูลเข้า	ข้อมูลออก
6	13 25 40	6	0:8
8		8	1:17
17		17	2:13 32
32		32	3:25 40
25		25	
13		13	
40		40	
1		1	
5		4	

ตัวอย่าง 4

ข้อมูลเข้า	ข้อมูลออก
10	10 20 30 40 100 90 80 70 60 50
50	10 100
40	10 20 30 40 50 60 70 80 90 100
30	
20	
10	
60	
70	
80	
90	
100	
3	
3 5 2	