

Ternary Tree Traversal

Time limit: 1 sec

memory limit: 512mb

ต้นไม้ตรีภาคคือต้นไม้ที่ปมแต่ละปมนั้นอาจจะมีลูกได้สามปม ได้แก่ ลูกซ้าย ลูกกลาง และ ลูกขวา กำหนดให้ การทำ preorder traversal และ inorder traversal ของปมในต้นไม้เขียนเป็นรหัสเทียมได้ดังนี้ โดยให้ left_child(x), center_child(x) และ right_child(x) คือ ลูกซ้าย ลูกกลาง และ ลูกขวา ของปม x ตามลำดับ

| Preorder | Inorder |
|--|--|
| <pre>preorder(node x) { if (x == NULL) return print(value(x)) preorder(left_child(x)) preorder(center_child(x)) preorder(right_child(x)) }</pre> | <pre>inorder(node x) { if (x == NULL) return inorder(left_child(x)) print(value(x)) inorder(center_child(x)) inorder(right_child(x)) }</pre> |

เรามีต้นไม้ตรีภาคขนาด n ปมอยู่ต้นหนึ่ง ซึ่งแต่ละปมนั้นกำกับด้วยจำนวนเต็ม 1 ตัว ซึ่งมีค่าตั้งแต่ 0 ถึง $n-1$ โดยที่แต่ละปมไม่ซ้ำกันเลย จากข้อมูลผลการทำงานของ preorder traversal และ inorder traversal ที่ปมรากของต้นไม้ต้นนี้ จงคำนวณว่ามีต้นไม้กี่ต้น (รวมต้นไม้ของเราด้วย) ที่มีผลการทำงาน preorder และ inorder ตรงกับที่กำหนดให้

Input

- บรรทัดแรกประกอบด้วยค่า n ($1 \leq n \leq 100$)
- บรรทัดที่สองประกอบด้วยจำนวนเต็ม n ตัวมีค่า 0 ถึง $n-1$ โดยไม่ซ้ำกันเลย เป็นผลการทำงานของ preorder traversal ของต้นไม้ของเรา
- บรรทัดที่สามประกอบด้วยจำนวนเต็ม n ตัวมีค่า 0 ถึง $n-1$ โดยไม่ซ้ำกันเลย เป็นผลการทำงานของ inorder traversal ของต้นไม้ของเรา

Output

ประกอบด้วยตัวเลขจำนวนเต็ม 1 ตัวซึ่งระบุจำนวนรูปแบบของต้นไม้ที่เป็นไปได้ทั้งหมด เนื่องจากค่าที่แสดงผลอาจจะมีค่าสูงมาก ให้แสดงเฉพาะจำนวนหารเอาเศษด้วย (10^9+7)

Example

| Input | Output |
|-----------------|--------|
| 1 0 0 | 1 |
| 2 1 0 1 0 | 2 |

| | |
|---------------------------------|----|
| 2 1 0 0 1 | 1 |
| 4 0 1 2 3 1 0 2 3 | 5 |
| 6 5 4 0 1 2 3 4 5 1 0 2 3 | 14 |