

List Block Swap

(1 sec, 512mb)

จงเพิ่มบริการ void CP::list<T>::block_swap(iterator a1, iterator a2, iterator b1, iterator b2) ให้กับ CP::list โดยฟังก์ชันนี้จะ “สลับข้อมูลทั้งช่วง” ของข้อมูลตั้งแต่ a1 ถึง “ก่อน a2” กับข้อมูลตั้งแต่ b1 ถึง “ก่อน b2”

ตัวอย่างเช่น หากใน list มีข้อมูลเป็น {10, 20, 30, 40, 50, 60, 70, 80, 90} โดยให้ตัวชี้ท้ายสุดในรายการคือข้อมูลตัวแรกสุดใน list และ a1, a2, b1, b2 คือ iterator ที่ชี้ข้อมูลที่มีค่า 20, 50, 80 และ 90 ตามลำดับ ผลจากการเรียก block_swap(a1,a2,b1,b2) จะทำให้ list นี้มีค่าเป็น {10, 80, 50, 60, 70, 20, 30, 40, 90} (ข้อมูลสีเขียวและสีเหลืองคือช่วงข้อมูลที่จะต้องโดนสลับ ส่วนข้อมูลที่ขีดเส้นใต้คือ a2 และ b2)

รับประกันว่าการเรียกฟังก์ชันนี้ ตำแหน่งของ a2 จะไม่อยู่ก่อน a1 และ b2 จะไม่อยู่ก่อน b1 และรับประกันว่าข้อมูลช่วงที่ต้องสลับนั้นจะไม่ซ้อนทับกันแน่ ๆ

ข้อบังคับ

- ในโจทย์ข้อนี้ ห้ามทำการสร้างปมของ list ใหม่ (ซึ่งหมายความว่าไม่สามารถใช้วิธีสร้าง list ใหม่ แล้วค่อย ๆ เติมปมเข้าไปได้) แต่อย่างไรก็ตาม 70% ของ testcase จะอนุญาตให้สร้างปมของ list ใหม่ได้
 - โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ Code::Blocks ให้ ซึ่งในไฟล์โปรเจกต์ดังกล่าวจะมีไฟล์ list.h, main.cpp และ student.h อยู่ ให้หিসิตเขียน code เพิ่มเติมลงในไฟล์ student.h เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ student.h เท่านั้น
 - ในไฟล์ student.h ดังกล่าวจะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือคีย์บอร์ดหรือไฟล์ใด ๆ
 - หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ main.cpp
- ** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจกต์เริ่มต้นแต่จะทำการทดสอบในลักษณะเดียวกัน ****

คำอธิบายฟังก์ชัน main

main() จะอ่านข้อมูลมา 2 บรรทัด คือ

- บรรทัดแรกประกอบด้วยจำนวนเต็ม N และ i1, i2, j1, j2 ซึ่งระบุจำนวนข้อมูลใน list และหมายเลขตำแหน่งของ a1, a2, b1, b2 ตามลำดับ (ตำแหน่งแรกสุดคือ 0)
- บรรทัดที่สองประกอบด้วยจำนวนเต็ม N ตัวคือข้อมูลที่จะใส่เข้าไปใน CP::list ตามลำดับ หลังจากนั้น main จะเรียก block_swap(a1,a2,b1,b2) และ ทำการพิมพ์ข้อมูลทั้งหมดใน list ออกมา

ชุดข้อมูลทดสอบ

ให้ i1, i2, j1, j2 คือ ตัวเลขที่บอกตำแหน่งของ iterator a1, a2, b1, b2 ตามลำดับ

- 20% รับประกันว่า $i1 + 1 == i2$ และ $j1 + 1 == j2$ และ $i1 < i2 < j1 < j2$ และสามารถสร้างปมใหม่ได้
- 30% รับประกันว่า $i1 < i2 < j1 < j2$ และสามารถสร้างปมใหม่ได้ และ $i2 - i1 \leq 100$ และ $j2 - j1 \leq 100$
- 20% รับประกันว่า $i2 < j1$ และสามารถสร้างปมใหม่ได้
- 30% ไม่มีเงื่อนไขอื่น ๆ

ตัวอย่างเพิ่มเติม

ให้ list มีข้อมูลเป็น {1, 2, 3, 4, 5, 6, 7, 8, 9} และเมื่อ i1, i2, j1, j2 มีค่าดังตารางต่อไปนี้ list ที่เป็นผลลัพธ์ของการเรียก block_swap(a1,a2,b1,b2) จะมีค่าเป็นดังนี้

i1	i2	j1	j2	ผลลัพธ์
2	3	5	6	{1, 2, 6, 4, 5, 3, 7, 8, 9}
5	6	2	3	{1, 2, 6, 4, 5, 3, 7, 8, 9}
2	2	5	6	{1, 2, 6, 3, 4, 5, 7, 8, 9}
2	2	5	5	{1, 2, 3, 4, 5, 6, 7, 8, 9}
2	2	2	5	{1, 2, 3, 4, 5, 6, 7, 8, 9}
0	5	5	9	{6, 7, 8, 9, 1, 2, 3, 4, 5}
5	9	0	5	{6, 7, 8, 9, 1, 2, 3, 4, 5}