Queue Insert Shuffle

(1 sec, 512mb)

ให้เขียนโปรแกรมเพื่อเพิ่มบริการ void CP::queue::insert_shuffle(std::vector<T> items, size_t pos) ให้กับ CP::queue โดยฟังก์ชันนี้จะทำหน้าที่แทรกข้อมูลใน items เริ่มที่ตำแหน่ง pos นับจากหัวของคิว ให้สลับกันไปมากับข้อมูลเดิมที่มีอยู่แล้วในคิว หากข้อมูลเดิมในคิวมีไม่มากพอให้สลับต่อแล้ว ก็ให้แทรกข้อมูลที่เหลือใน items ต่อท้ายคิว

ตัวอย่างเช่น เริ่มต้นให้คิวมีข้อมูลเป็น [1, 3, 5, 7, 9] และให้หัวคิวอยู่ที่ตำแหน่งซ้ายสุด หากมีการเรียกใช้ insert_shuffle โดยมี items = [10, 20, 30, 40] และ pos = 2 จะทำให้ได้คิวที่มีข้อมูลเป็น [1, 3, 10, 5, 20, 7, 30, 9, 40]

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์ตั้งต้นมาให้ ซึ่งประกอบด้วยไฟล์ queue.h, main.cpp และ student.h อยู่ ให้นิสิตเขียน code เพิ่มเติมลงในไฟล์ student.h เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ student.h เท่านั้น
- ไฟล์ student.h จะต้องไม่ทำการอ่านเขียนข้อมูลใดๆ ไปยังหน้าจอหรือคีย์บอร์ดหรือไฟล์ใดๆ
- หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ main.cpp
  **** main ที่ใช้จริงใน grader นั้นจะต่างจาก main ที่ได้รับในไฟล์โปรเจ็กต์เริ่มต้น แต่จะทำการทดสอบในลักษณะเดียวกัน ****

คำอธิบายฟังก์ชัน main

main() จะอ่านข้อมูล 3 บรรทัดคือ

- บรรทัดแรก ประกอบด้วยจำนวนเต็มสามตัวคือ N, M และ pos ซึ่งระบุถึงจำนวนข้อมูลเดิมในคิว, จำนวนข้อมูลใน items และ ตำแหน่งเริ่มต้นที่ต้องการจะแทรกข้อมูล (1 <= N, M <= 500,000, 0 <= pos <= N)
- บรรทัดที่สอง ประกอบด้วยจำนวนเต็ม N ตัวคือ ข้อมูลแต่ละตัวที่อยู่ในคิว โดยเริ่มจากหัวคิวไปยังท้ายคิว
- บรรทัดที่สาม ประกอบด้วยจำนวนเต็ม M ตัวคือ ข้อมูลแต่ละตัวที่อยู่ใน items

หลังจากนั้น main() จะเรียกใช้บริการ insert_shuffle และทำการแสดงผลข้อมูลทั้งหมดในคิวออกมา

ข้อมูลชุดทดสอบ

- 5% ข้อมูลใน insert_shuffle มีเพียงตัวเดียว
- 10% mFront = 0 และ เมื่อเรียกใช้ insert_shuffle จะไม่มีข้อมูลที่ต้องการแทรกเกินออกไปที่ท้ายคิว
- 15% เมื่อเรียกใช้ insert_shuffle จะไม่มีข้อมูลที่ต้องการแทรกเกินออกไปที่ท้ายคิว
- 20% จำนวนข้อมูลหลังถูกแทรกจะมีค่า**มากกว่า** mCap เดิมของคิว
- 50% ไม่มีข้อกำหนดพิเศษอื่นใด

(มีตัวอย่างอยู่หน้าถัดไป)

ตัวอย่างการทำงานของ main

ข้อมูลนำเข้า	ข้อมูลส่งออก
5 4 2 1 3 5 7 9 10 20 30 40	1 3 10 5 20 7 30 9 40
5 5 0 10 20 30 40 50 5 4 3 2 1	5 10 4 20 3 30 2 40 1 50
10 2 10 1 3 5 7 11 13 17 19 23 29 3 4	1 3 5 7 11 13 17 19 23 29 3 4