Queue Block Insert

(1 sec, 512mb)

จงเพิ่มบริการ void CP::queue::block_insert(size_t p, size_t m, const T &element) ให้กับ CP::queue โดยฟังก์ชันนี้จะ “แทรก” ข้อมูล element เป็นจำนวน m ตัว เข้าไปใน CP::queue ที่เรียกใช้ฟังก์ชันนี้ให้เป็นข้อมูลในลำดับที่ p ถึง p+m-1 โดยกำหนดให้ข้อมูลในลำดับที่ p หมายถึงข้อมูลที่อยู่ถัดจากหัวคิว (ข้อมูลที่ได้จากการเรียก front()) ไปเป็นจำนวน p ช่อง จึงได้ว่าข้อมูลลำดับที่ 0 คือข้อมูลที่หัวคิว ข้อมูลลำดับที่ 1 คือข้อมูลที่อยู่ถัดจากหัวคิว (กล่าวอีกนัยหนึ่งคือ ข้อมูลในลำดับที่ p คือ ข้อมูลที่จะอยู่ที่หัวคิว หากเราทำการ pop() ไป p ครั้ง)

ตัวอย่างเช่น หากให้ queue เรามีข้อมูลอยู่ภายในเป็น {10, 20, 30, 40, 50, 60} โดยที่ด้านซ้ายสุดเป็นหัวคิว (กล่าวคือ front() คือค่า 10) แล้วเราเรียก block_insert(2,5,-1) จะทำให้ queue นี้มีข้อมูลเป็น {10, 20, -1, -1, -1, -1, -1, 30, 40, 50, 60}

รับประกันว่า $0 \leq p \leq$ size()

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 จะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือคีย์บอร์ดหรือไฟล์ใด ๆ
- หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ main.cpp

**** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจ็กต์เริ่มต้นแต่จะทำการทดสอบในลักษณะเดียวกัน ****

คำอธิบายฟังก์ชัน main

main() จะทดลองใช้งาน CP::queue โดย main จะสร้าง CP::queue<int> ชื่อ q มาแล้วอ่านคำสั่งทีละบรรทัด ซึ่งแต่ละบรรทัดจะเริ่มต้นด้วย string ที่ระบุคำสั่ง และอาจจะตามด้วยค่าต่าง ๆ ที่จำเป็นสำหรับคำสั่งนั้น โดยมีรูปแบบของแต่ละคำสั่งดังต่อไปนี้ (ให้ X, e หมายถึงค่าประเภท int ใด ๆ)

คำสั่ง	คำอธิบาย
s	พิมพ์ขนาดของ q
u X	เรียก q.push(X)
o	เรียก q.pop()
f	พิมพ์ค่าของ q.front()
b	พิมพ์ค่าของ q.back()
i p m e	เรียก q.block_insert(p,m,e)
q	จบการทำงาน

ชุดข้อมูลทดสอบ

- 15% m = 1 เสมอ และรับประกันว่าเมื่อเรียก block_insert นั้นจะมีค่า mFront + mSize < mCap
- 15% m = 1
- 20% p ≤ 0.2 * mSize (แทรกในตำแหน่งต้น ๆ ของ queue เสมอ)
- 20% p ≥ 0.8 * mSize (แทรกในตำแหน่งท้าย ๆ ของ queue เสมอ)
- 30% ไม่มีข้อจำกัดอื่นใด (มีทั้งแทรกต้นๆ แทรกท้าย ๆ และแทรกตำแหน่งอื่น ๆ)

คำแนะนำ

เราสามารถแก้ไข queue อย่างไรก็ได้ ขอให้ข้อมูลใน queue มีค่าตามลำดับที่ถูกต้อง (เช่น สามารถแก้ไข mFront, mSize หรืออื่น ๆ ได้)

ตัวอย่างการทำงานของ main

ข้อมูลนำเข้า	ข้อมูลส่งออก
u 10 u 20 i 1 3 -1 f o f s b q	10 // ค่า front ของ {10, -1, -1, -1, 20} -1 // ค่า front ของ {-1, -1, -1, 20} 4 // ค่า size 20 // ค่า back
u 69 i 0 8 80 s o s f i 5 6 18 f o u 65 b i 2 10 16 f u 56 q	9 // ค่า size 8 // ค่า size หลังเรียก pop 80 // ค่า front 80 // ค่า front 65 // ค่า back 80 // ค่า front
i 0 2 5 f o b i 1 1 3 b o f q	5 // ค่า front ของ {5, 5} 5 // ค่า back ของ {5} 3 // ค่า back ของ {5, 3} 3 // ค่า front ของ {3}