

# Vector Range Erase

(1 sec, 512mb)

จงเพิ่มบริการให้กับ `CP::vector` เพื่อใช้สำหรับการลบข้อมูลเป็นช่วง ๆ แบบหลาย ๆ ช่วง กล่าวคือ จงเพิ่มฟังก์ชัน `range_erase(std::vector<pair<iterator,iterator>> ranges)` โดยที่ `iterator` คือ `CP::vector<T>::iterator` และเราต้องการลบข้อมูลทุกตัวที่อยู่ตั้งแต่ `range[i].first` จนถึงก่อน `range[i].second` สำหรับค่า `i` ใด ๆ ที่  $0 \leq i < \text{ranges.size}()$  พร้อมกัน

รับประกันว่าเมื่อเรียกฟังก์ชันนี้ `ranges[i].first` และ `ranges[i].second` นั้นจะมีค่าที่เป็นไปได้ตั้งแต่ `begin()` จนถึง `end()` ของ `CP::vector` แน่แน่นอน และ `ranges[i].first < ranges[i].second`

## คำแนะนำ

- เป็นไปได้ที่ `ranges[i]` และ `range[j]` นั้นอาจจะระบุถึงช่วงที่ซ้อนทับกัน
- อย่าลืมว่าการเรียก `erase()` นั้นจะมีการเลื่อนข้อมูล

## คำอธิบายฟังก์ชัน `main()`

`main` จะอ่านข้อมูลมาหลายบรรทัด ตามรูปแบบนี้

- บรรทัดแรกประกอบด้วยจำนวนเต็ม `n` และ `m` ( $1 \leq n, m \leq 500,000$ )
  - โปรแกรมจะสร้าง `CP::vector<int>` ขนาด `n` ช่อง คือ `v` และให้ให้ `v[i] = i`
- หลังจากนั้นอีก `m` บรรทัดจะเป็นข้อมูลของ `ranges` โดยที่แต่ละบรรทัด ซึ่งประกอบด้วยจำนวนเต็ม 2 ตัวคือ `a` และ `b` จะระบุค่าของ `ranges[i]` โดยให้ `ranges[i].first` คือ `v.begin() + a` และ `ranges[i].second` คือ `v.begin() + b`
- หลังจากนั้น `main` จะเรียก `v.range_erase(ranges)` แล้วทำการพิมพ์ค่าทั้งหมดของ `v` ออกมา

## ชุดข้อมูลทดสอบ

- 20%  $n, m \leq 1,000$  และ `ranges[i].second <= ranges[i+1].first` เสมอ
- 30%  $n, m \leq 1,000$
- 50% ไม่มีเงื่อนไขอื่นใด

## ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ Code::Blocks ให้ ซึ่งในไฟล์โปรเจ็คดังกล่าวจะมีไฟล์ `vector.h`, `main.cpp` และ `student.h` อยู่ ให้นิสิตเขียน 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` ที่ได้รับในไฟล์โปรเจ็คเริ่มต้น \*\*

## ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
10 3	2
0 2	3
4 5	5
9 10	6
	7
	8
20 8	0
2 4	1
15 20	4
13 14	5
8 9	11
6 11	14
10 11	
12 14	
18 19	

## คำอธิบายตัวอย่างชุดที่ 2

สำหรับตัวอย่างข้อมูลนำเข้าที่ 2 จะเห็นว่าช่วงที่ทับซ้อนอย่าง (6, 11), (10, 11), (8, 9) จะนับเป็นการลบครั้งเดียวคือตั้งแต่ช่องที่ 6 ถึง 10 (**ไม่นับ 11**) ดังนั้นค่าที่ถูกลบไปจึงมีดังนี้

- 2, 3 (จากช่วงที่ 1)
- 6, 7, 8, 9, 10 (จากช่วงที่ 4, 5, 6)
- 12, 13 (จากช่วงที่ 3, 7)
- 15, 16, 17, 18, 19, 20 (จากช่วงที่ 2, 8)