Hash Next Cost

(1 sec, 512mb)

CP::unordered_map เป็น Hash Table ที่แก้ปัญหาการชนกันด้วยวิธี Separate Chaining โดยที่ iterator ของ CP::unordered_map นั้นจะต้องจำทั้งตำแหน่งของข้อมูลใน Bucket และตำแหน่งของ Bucket ในตาราง Hash

การเรียกใช้ operator++ หรือ operator-- ของ iterator นั้น ในบางครั้งจะเป็นการวิ่งจาก Bucket หนึ่งไปยังอีก Bucket หนึ่ง ซึ่งอาจจะไม่ได้ใช้เวลาคงที่ก็เป็นได้ ขอให้พิจารณาจากฟังก์ชัน to_next_data ในคลาส CP::unordered_map

จงเพิ่มบริการ size_t CP::unordered_map<T> next_cost(iterator it) คำนวณว่า การเรียกใช้ operator++ ของ it ที่ให้มาเป็น argument ของฟังก์ชัน next_cost นั้น ต้องมีการขยับตำแหน่งกี่ครั้งถึงจะไปยังข้อมูลตัวถัดไปใน hash (หรือไปยัง end())

ขอให้พิจารณาตัวอย่างต่อไปนี้

- หาก Hash Table ของเรามีข้อมูลสองตัว และทั้งสองตัวถูก Hash ลงช่องเดียวกัน และให้ iterator it ชี้ไปยังตำแหน่งของข้อมูลตัวแรก การเรียก next_cost(it) นั้นจะต้องคืนค่า 1 เนื่องจากเราต้องขยับตำแหน่งครั้งเดียว คือ การขยับจากตัวแรกใน bucket ไปยังตัวที่สองใน bucket
- หาก Hash Table ของเรามีข้อมูลสองตัว โดยตัวแรกถูก Hash อยู่ใน Bucket ในช่องหมายเลข 5 และตัวที่สองถูก Hash อยู่ใน Bucket ในช่องหมายเลข 13 และให้ iterator it ชี้ไปยังตำแหน่งของข้อมูลตัวแรก การเรียก next_cost(it) นั้นจะต้องคืนค่า 9 เนื่องจากเราต้องขยับตำแหน่ง 9 ครั้ง คือ การขยับจากตัวแรกใน bucket ไปยัง end() ของ bucket นั้น ซึ่งทำให้ต้องขยับไปยัง Bucket ถัดไป (Bucket ที่ 6) รวมเป็นสองครั้ง และต้องขยับไปยัง Bucket ถัดไปอีกเรื่อย ๆ (Bucket ที่ 7, 8, 9, ..., 13) อีก 7 ครั้งรวมเป็น 9 ครั้ง
- หาก Hash Table ของเรามีข้อมูลสองตัว โดยตัวแรกถูก Hash อยู่ใน Bucket ในช่องหมายเลข 5 และตัวที่สองถูก Hash อยู่ใน Bucket ในช่องหมายเลข 13 และให้ตาราง Hash มีขนาดเป็น 15 ช่อง (ช่องหมายเลข 0 ถึง 14) และให้ iterator it ชี้ไปยังตำแหน่งของข้อมูลตัวที่สอง (Bucket หมายเลข 13) การเรียก next_cost(it) นั้นจะต้องคืนค่า 3 เนื่องจากเราต้องขยับตำแหน่ง 3 ครั้งถึงจะถึงตำแหน่ง end() คือ
    1. การขยับจากตัวแรกใน bucket ไปยัง end() ของ bucket นั้น
    2. การขยับไปยัง Bucket ถัดไป (Bucket ที่ 14)
    3. การขยับไปยัง end() ของตาราง Hash ของเรา

    รวมเป็นการขยับทั้งหมด 3 ครั้ง

คำอธิบายฟังก์ชัน main

main() จะสร้าง CP::unordered_map<int,int> h จากข้อมูลที่ได้รับมา โดย main จะรับข้อมูลสามบรรทัด

- บรรทัดแรกรับจำนวนเต็ม n ซึ่งจะบุจำนวนข้อมูลใน h
- บรรทัดที่สองประกอบด้วยจำนวนเต็ม n ตัว ซึ่งคือ key ของข้อมูลใน h
- บรรทัดที่สามประกอบด้วยจำนวนเต็ม a (รับประกันว่า a <= n)

หลังจาก main จะสร้าง it ให้ชี้ไปยัง h.begin() แล้วทำการเรียก it++ เป็นจำนวน a ครั้ง แล้วจึงเรียก next_cost(it) พร้อมทั้งแสดงผลลัพธ์ของการเรียก

ชุดข้อมูลทดสอบ

- 5% h ที่เรียกมีข้อมูล 1 ตัว และข้อมูลที่เก็บใน h เป็น int
- 10% h ที่เรียกมีข้อมูล 2 ตัว และข้อมูลที่เก็บใน h เป็น int
- 10% h ที่เรียกมีข้อมูล 0 ตัว และข้อมูลที่เก็บใน h เป็น int
- 20% h ที่เรียกมีข้อมูล 10 ตัว และข้อมูลที่เก็บใน h เป็น int และ a = n
- 10% h ที่เรียกมีข้อมูล 100 ตัว และ a = n
- 45% ไม่มีข้อบังคับอื่นใด

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ Code::Blocks ให้ ซึ่งในไฟล์โปรเจ็คดังกล่าวจะมีไฟล์ unordered_map.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 ที่ได้รับในไฟล์โปรเจ็กต์เริ่มต้นแต่จะทำการทดสอบในลักษณะเดียวกัน **

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
1 0 0	12 คำอธิบาย: n = 1 0 a = 0 ---------- bucket count = 11 bucket #0 [0]
2 0 4 0	5 คำอธิบาย: n = 2 0 4 (ข้อมูลสองตัวอยู่คนละช่อง) a = 0 ---------- bucket count = 11 bucket #0 [0] bucket #4 [4]

2
0 4
1
	8
	
	คำอธิบาย:
	n = 2
	0 4 (ข้อมูลสองตัวอยู่คนละช่อง)
	a = 1
	---------- 	
	bucket count = 11
	bucket #0
	  [0]
	 bucket #4
	  [4]

2
0 4
2
	0
	
	คำอธิบาย:
	n = 2
	0 4 (ข้อมูลสองตัวอยู่คนละช่อง)
	a = 2
	---------- 	
	bucket count = 11
	bucket #0
	  [0]
	 bucket #4
	  [4]

2
0 5
0
	1
	
	คำอธิบาย:
	n = 2
	0 5 (hash ลงช่องเดียวกัน)
	a = 0
	---------- 	
	bucket count = 11
	bucket #0
	  [0]
	  [5]

2 0 5 1	12 คำอธิบาย: n = 2 0 5 (hash ลงช่องเดียวกัน) a = 1

	---------- 	
	bucket count = 11
	bucket #0
	  [0]
	  [5]

5
0 1 2 3 4
0
	2
	
	คำอธิบาย:
	n = 5
	0 1 2 3 4
	a = 0
	---------- 	
	bucket count = 11
	bucket #0
	  [0]
	 bucket #1
	  [1]
	 bucket #2
	  [2]
	 bucket #3
	  [3]
	 bucket #4
	  [4]