

Lecture 11

เอกสารประกอบการบรรยาย Data Structure

เรื่อง ตาราง แฮช (Hash Table)

เนื้อหาการเรียนรู้

- ตารางการเข้าถึงข้อมูลแบบตรง
- ตารางแฮช
- การชนกันของข้อมูล
- การแก้ปัญหการชนกันของข้อมูล
- วิธีการสร้างฟังก์ชันแฮช

จุดประสงค์การเรียนรู้

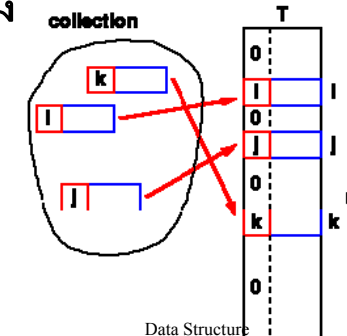
1. เพื่อให้ นักศึกษาทราบถึงวิธีการเข้าถึงข้อมูลแบบตรง
2. เพื่อให้ นักศึกษาทราบวิธีการสร้างฟังก์ชันแฮช
3. เพื่อให้ นักศึกษาทราบวิธีการตรวจสอบข้อมูล
4. เพื่อให้ นักศึกษาสามารถแก้ไขปัญหการชนกันของข้อมูล
5. เพื่อให้ นักศึกษาทราบวิธีการชนกันของข้อมูล

ตารางข้อมูลแบบตรง (Direct-address table)

สมมติว่ามีการกำหนดให้คีย์มาจากเอกภพสัมพัทธ์

$$U = \{0, 1, \dots, m-1\}$$

การแก้ไขปัญหาคือใช้ตาราง $T[0..m-1]$ การสร้างดัชนีโดยคีย์เพื่อใช้ในการ เชื่อมโยงข้อมูล เข้าด้วยกันเพื่อเก็บข้อมูลเข้าในตำแหน่งที่ถูกต้อง



ตารางแฮช (Hash Tables)

เมื่อขนาดของเอกภพสัมพัทธ์เพิ่มมากขึ้น ตามหลักการยังคงสามารถทำงานได้ แต่ขนาดของตาราง T จะมีผลกระทบ

ทางแก้ปัญหาคือต้องหาวิธีการจับคู่คีย์ให้มีช่วงกว้างที่เล็กลงโดยเรียกวิธีการนี้ว่า

ฟังก์ชันแฮช (Hash Function)

ผลลัพธ์ที่ได้เรียกว่า

ตารางแฮช (Hash Table)

ตารางแฮช (Hash Tables)

การเข้าถึงข้อมูลโดยตรง กำหนด ให้ k เป็นคีย์ ถูกจัดเก็บอยู่ใน ช่อง k ด้วยการทำแฮชด้วยพื้นฐาน การจัดเก็บในช่องที่ $h(k)$ โดยใช้ฟังก์ชัน h เพื่อคำนวณหาช่องของคีย์ โดยการจับคู่กับเอกภพสัมพัทธ์ U ในตาราง T

$$h: U \rightarrow \{0, 1, \dots, m-1\}$$

ตารางแฮช (Hash Tables)

ฟังก์ชัน แฮช จะทำงานแบบสุม ตัวอย่างเช่น

$h(k) = k \bmod m$ เมื่อ m เป็นค่าหลัก (prime number) จำนวนช่อง (Slot)

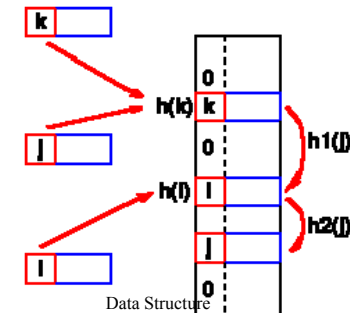
แนวคิดการจัดเก็บ ค่าคีย์ของ k ในตำแหน่ง ที่ $T[h(k)]$

เมื่อ $k \in U$, $h(k) \in [0..m-1]$

แนวคิดหลัก คือ ลด ขนาดอะเรย์ของดัชนี

การชนกันของข้อมูล (Collision)

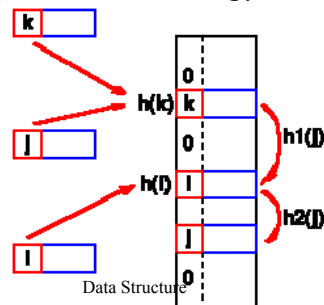
การที่แทรกคีย์ในตาราง ที่จัดเก็บนั้นมีโอกาสที่คีย์ที่ถูกสร้างจากฟังก์ชัน ในช่องเดียวกัน อย่างไรก็ตามการเกิดการชนกันก็ยังคงต้องมีอย่างน้อยหนึ่งครั้ง



การชนกันของข้อมูล (Collision)

จากรูป ฟังก์ชัน h การจับคู่ระหว่าง k กับ j ในช่องทางเดียวกัน ซึ่ง อาจจะชนกัน

สามารถแบ่งวิธีการในการรองรับการชนกันของตารางแฮชคือ การทำแบ่งห่วงโซ่ (Chaining) แบบการเปิดที่อยู่ (Open Addressing)



การแก้ไขปัญหาคollisionของข้อมูล แบบห่วงโซ่(Chaining)

$$h(k_1) = h(k_n) \text{ and } h(k_5) = h(k_2) = h(k_7)$$

keys: 5, 28, 19, 15, 20, 33, 12, 17, 10

slots: 9

hash function = $h(k) = k \bmod 9$

$$h(5) = 5 \bmod 9 = 5$$

$$h(28) = 28 \bmod 9 = 1$$

$$h(19) = 19 \bmod 9 = 1$$

$$h(15) = 15 \bmod 9 = 6$$

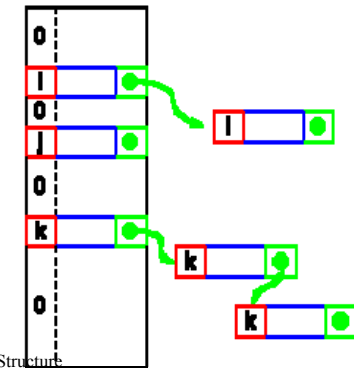
$$h(20) = 20 \bmod 9 = 2$$

$$h(33) = 33 \bmod 9 = 6$$

$$h(12) = 12 \bmod 9 = 3$$

$$h(17) = 17 \bmod 9 = 8$$

$$h(10) = 10 \bmod 9 = 1$$



การแก้ไขปัญหาคollisionของข้อมูล แบบห่วงโซ่(Chaining)

1. กรณีที่เลวร้ายที่สุด ในการแทรกข้อมูลคือ $O(1)$
2. การลบสมาชิก สามารถทำได้ด้วยเวลาที่น้อยที่สุดของ $O(1)$

ทางปฏิบัติ ใช้เทคนิค ฮิวริสติก (Heuristic) ในการสร้างฟังก์ชันแฮช แนวทางหนึ่งที่ดีคือ การแปลงค่าของข้อมูลที่มีอยู่แล้วด้วยข้อมูลที่มีอยู่ (วิธีการหาร: Division method)

ฟังก์ชันแฮช คือการกำหนดค่าคีย์ที่เกิดขึ้นในเอกภพสัมพัทธ์จากตัวเลขธรรมชาติ

วิธีการสร้างฟังก์ชันแฮช (Method for Creating Function)

1. วิธีการหาร (The Division Method)
2. วิธีการคูณ (The Multiplication Method)
3. วิธีทั่วไป (Universal hashing)

วิธีการหาร (The Division Method)

การจับคู่คีย์ k ในช่องของ m โดยนำเศษที่เหลือของ k จากการหารด้วย m ด้วยฟังก์ชันคือ

$$h(k) = k \bmod m.$$

ตัวอย่างเช่น :

ถ้าตารางมีขนาด $m = 12$ ช่อง

คีย์ $k = 100$

ดังนั้น จะได้ $h(100) = 100 \bmod 12$
 $= 4$

วิธีการคูณ (The Multiplication Method)

ประกอบด้วย 2 ขั้นตอนดังนี้

ขั้นที่ 1 : คูณ k ด้วยค่าคงที่ A , $0 < A < 1$
ด้วยการขยาย เศษส่วนของ kA

ขั้นที่ 2 : การคูณ kA ด้วย m

$$h(k) = \lceil m(kA \bmod 1) \rceil$$

เมื่อ " $kA \bmod 1$ " หมายถึง เศษส่วนของ kA , นั่นคือ $kA - \lfloor kA \rfloor$.

ประโยชน์ของวิธีนี้ คือ ค่าของ m จะไม่วิกฤต และสามารถดำเนินการในเครื่องคอมพิวเตอร์ส่วนมากได้

วิธีทั่วไป (Universal hashing)

Open Addressing

ฟังก์ชันแฮช คือ

$h: \mathcal{V} \{0, 1, \dots, m-1\} \rightarrow \{0, 1, \dots, m-1\}$

ลำดับในการตรวจสอบ (probe sequence) คือ

$\langle h(k, 0), h(k, 1), \dots, h(k, m-1) \rangle$

เทคนิคลำดับของการตรวจสอบ

1. การตรวจสอบเชิงเส้น (Linear Probing)
2. การตรวจสอบด้วยสมการกำลังสอง (Quadratic Probing)
3. การสร้างฟังก์ชันแฮชแบบสองเท่า (Double Hashing)

เทคนิคลำดับของการตรวจสอบ

1. การตรวจสอบเชิงเส้น (Linear Probing)

รูปแบบของ ฟังก์ชันคือ

$$h(k, i) = (h'(k) + i) \bmod m$$

เมื่อ $i = 0, 1, 2, \dots, m-1$

h' คือ auxiliary ของฟังก์ชันแฮช

เทคนิคลำดับของการตรวจสอบ

2. การตรวจสอบด้วยสมการกำลังสอง (Quadratic Probing)

รูปแบบของ ฟังก์ชันคือ

$$h(k, i) = (h'(k) + c_1i + c_2i^2) \bmod m$$

เมื่อ $i = 0, 1, 2, \dots, m-1$

h' คือ auxiliary ของฟังก์ชันแฮช

$c_1 + c_2 \neq 0$ เป็นค่าคงที่แบบ auxiliary

เทคนิคลำดับของการตรวจสอบ

3. การสร้างฟังก์ชันแฮชแบบสองเท่า (Double Hashing)

รูปแบบของ ฟังก์ชันคือ

$$h(k, i) = (h_1 + 9k + ih_2(k)) \bmod m$$

เมื่อ h_1 และ h_2 เป็น auxiliary ของฟังก์ชัน

ค่า k เป็นค่าเริ่มต้นของ ตำแหน่งการตรวจสอบ และค่า *offset*

แบบฝึกหัด

1. จงจัดชุดข้อมูลลงในตาราง Hash Table

13,25,63,69,45,78,59,17

ขนาด 9 ช่อง โดยการแก้ปัญหาการชนกัน ในลักษณะแบบเซน

2. จงอธิบายวิธีการแก้ไขปัญหาการชนกันของข้อมูล

3. วิธีในการสร้าง Hash Function มีวิธีใดบ้าง