คำถามที่ 10

1. ข้อใดไม่เกี่ยวข้องกับเทคนิคการค้นหาข้อมูล
2. การค้นหาแบบลัดคิว (Sequential Search) คืออะไร
3. ข้อใดคือข้อดีของการค้นหาแบบลัดคิว
4. การค้นหาแบบเซนทinel (Sentinel Search) คืออะไร 5
5. ข้อใดเป็นข้อดีของการค้นหาแบบลัดคิว
6. ข้อใดคือข้อเสียของการค้นหาแบบลัดคิว
7. ข้อใดคือข้อดีของการค้นหาแบบลัดคิว
8. การค้นหาแบบความน่าจะเป็น (Probability Search) คืออะไร
9. การค้นหาแบบรายชื่อเรียง (Ordered List Search) 10
10. ข้อใดคือข้อดีของการค้นหาแบบรายชื่อเรียง
11. การค้นหาแบบไบนารี (Binary Search) คืออะไร
12. ข้อใดคือข้อดีของการค้นหาแบบไบนารี
13. ข้อใดคือข้อเสียของการค้นหาแบบไบนารี
14. การค้นหาแบบรายชื่อเรียงที่เรียงตามลำดับคือการใช้ชื่ออะไร 15
15. การค้นหาแบบแฮช (Hashed Search) คืออะไร
16. ข้อใดคือข้อดีของการค้นหาแบบแฮช
17. ข้อใดคือข้อเสียของการค้นหาแบบแฮช
18. ข้อใดคือข้อดีของการค้นหาแบบแฮช
19. ข้อใดคือข้อเสียของการค้นหาแบบแฮช 20
20. การค้นหาแบบโมดูลาร์ (Modulo-Division Hashing)
21. ข้อใดคือข้อดีของการค้นหาแบบโมดูลาร์
22. ข้อใดคือข้อเสียของการค้นหาแบบโมดูลาร์
23. วิธีการแก้ปัญหาคollision คือ Collision Resolution คือ
24. ข้อใดคือวิธีการแก้ปัญหาคollision 25
25. ข้อใดคือวิธีการ Linear Probe
26. ข้อใดคือวิธีการ Linear Probe
27. ข้อใดคือวิธีการ Linear Probe
28. Chaining ในข้อใด คืออะไร
29. วิธีการแก้ปัญหาคollision คือวิธีการใด 30
30. การแก้ปัญหาคollision คือวิธีการใด



คำถามที่ 10

1. สมมติว่าเรามีอาร์เรย์ของตัวเลขที่ไม่เรียงลำดับ และเราต้องการหาค่าที่เฉพาะเจาะจงในอาร์เรย์นี้ เราสามารถใช้การค้นหาแบบง่าย ๆ ได้หรือไม่? ถ้าใช่ เราจะใช้วิธีการใด? ถ้าไม่ใช่ เราจะใช้วิธีการใด?

5

2. สมมติว่าเรามีอาร์เรย์ของตัวเลขที่ไม่เรียงลำดับ และเราต้องการหาค่าที่เฉพาะเจาะจงในอาร์เรย์นี้ เราสามารถใช้การค้นหาแบบง่าย ๆ ได้หรือไม่? ถ้าใช่ เราจะใช้วิธีการใด? ถ้าไม่ใช่ เราจะใช้วิธีการใด?

10

3. สมมติว่าเรามีอาร์เรย์ของตัวเลขที่ไม่เรียงลำดับ และเราต้องการหาค่าที่เฉพาะเจาะจงในอาร์เรย์นี้ เราสามารถใช้การค้นหาแบบง่าย ๆ ได้หรือไม่? ถ้าใช่ เราจะใช้วิธีการใด? ถ้าไม่ใช่ เราจะใช้วิธีการใด?

15

4. สมมติว่าเรามีอาร์เรย์ของตัวเลขที่ไม่เรียงลำดับ และเราต้องการหาค่าที่เฉพาะเจาะจงในอาร์เรย์นี้ เราสามารถใช้การค้นหาแบบง่าย ๆ ได้หรือไม่? ถ้าใช่ เราจะใช้วิธีการใด? ถ้าไม่ใช่ เราจะใช้วิธีการใด?

20

5. สมมติว่าเรามีอาร์เรย์ของตัวเลขที่ไม่เรียงลำดับ และเราต้องการหาค่าที่เฉพาะเจาะจงในอาร์เรย์นี้ เราสามารถใช้การค้นหาแบบง่าย ๆ ได้หรือไม่? ถ้าใช่ เราจะใช้วิธีการใด? ถ้าไม่ใช่ เราจะใช้วิธีการใด?

Algorithm seqSearch (list, last, target, locn)

Locate the target in an unordered list of elements.

Pre list must contain at least one item

last is index to last element in the list

target contains the data to be located

locn is address of index in calling algorithm

Post if found : index stored in locn & found true

if not found : last stored in locn & found false

Return found true or false

1 set looken to 0

25

30



(ต่อไป) 2. loop (looker < last AND target not equal list[looker])

1 increment looker

3 end loop

4 set locn to looker

5 if (target equal list[looker])

1 set found to true

6 else

1 set found to false

7 end if

8 return found

end seqSearch

6. ตอน ตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target

2. loop (looker < last AND target not equal list[looker])

1 increment looker

สามารถดูตัวอย่างโค้ดที่แนบมาได้อีกด้วย สำหรับ การตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target นั้นใช้เวลา $O(n)$ นั่นหมายความว่าถ้าอาร์เรย์มีขนาด n เราจะต้องตรวจสอบทุกตัวในอาร์เรย์เพื่อหาค่าที่ตรงกับ target ซึ่งใช้เวลา $O(n)$

7. ตอน ตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target โดยไม่ต้องใช้ loop
ดูตัวอย่างโค้ดที่แนบมาได้อีกด้วย สำหรับ การตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target นั้นใช้เวลา $O(1)$ นั่นหมายความว่าถ้าอาร์เรย์มีขนาด n เราสามารถหาค่าที่ตรงกับ target ได้โดยไม่ต้องตรวจสอบทุกตัวในอาร์เรย์ ซึ่งใช้เวลา $O(1)$

8. ตอน ตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target (Probability Search) ซึ่งเป็นการสุ่มหาตำแหน่งที่มีค่าตรงกับ target โดยไม่ต้องใช้ loop
ดูตัวอย่างโค้ดที่แนบมาได้อีกด้วย สำหรับ การตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target นั้นใช้เวลา $O(1)$ นั่นหมายความว่าถ้าอาร์เรย์มีขนาด n เราสามารถหาค่าที่ตรงกับ target ได้โดยไม่ต้องตรวจสอบทุกตัวในอาร์เรย์ ซึ่งใช้เวลา $O(1)$

9. ตอน ตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target โดยไม่ต้องใช้ loop
ดูตัวอย่างโค้ดที่แนบมาได้อีกด้วย สำหรับ การตรวจสอบความยาวของอาร์เรย์และหาตำแหน่งที่มีค่าตรงกับ target นั้นใช้เวลา $O(1)$ นั่นหมายความว่าถ้าอาร์เรย์มีขนาด n เราสามารถหาค่าที่ตรงกับ target ได้โดยไม่ต้องตรวจสอบทุกตัวในอาร์เรย์ ซึ่งใช้เวลา $O(1)$



[illegible]

11. ตอน ทดสอบแบบฝึกหัดหน้า 141 ถึง 145 ของหนังสือเรียนภาษาอังกฤษ 5
ซึ่งมีผลต่อคะแนนของนักเรียน และนักเรียนต้องสอบผ่าน 9. ส่วน ด้านอื่น ด้านที่
ครูเห็นว่ามีความจำเป็นในด้านอื่น ๆ ของโรงเรียน (ซึ่งหมายถึงโรงเรียน)

12. นาย อรสาภากร อิ่มนวล กักตุนสินค้าเพื่อใช้ในโครงการช่วยเหลือผู้ประสบภัย

$$\text{mid} = \lfloor (\text{begin} + \text{end}) / 2 \rfloor$$

19. 0521

begin mid end

a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] a[8] a[9] a[10] a[11]

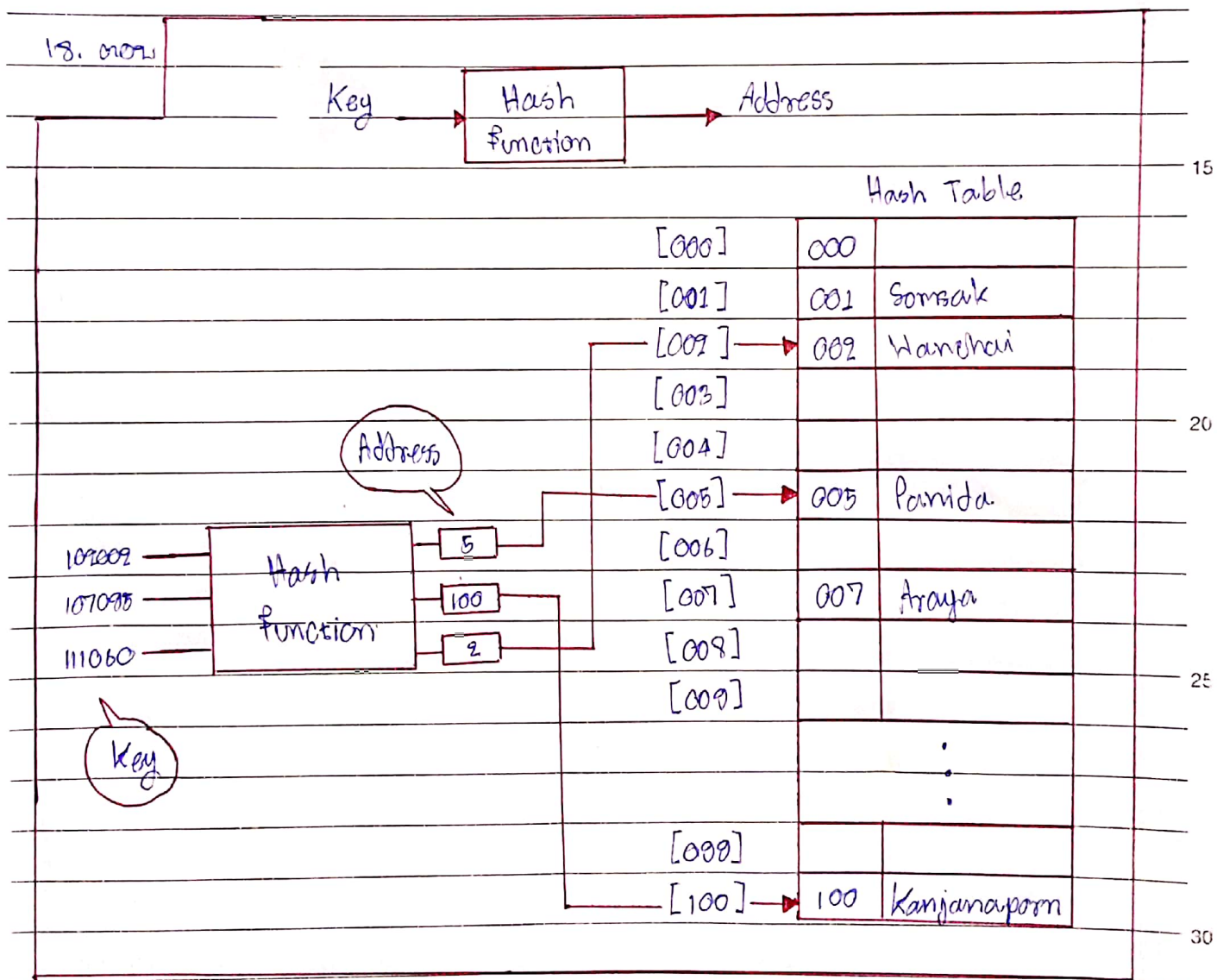
9	7	8	10	110	91	92	96	62	57	81	91
---	---	---	----	-----	----	----	----	----	----	----	----

92 > 91



17. ตาราง	ขนาดข้อมูล	จำนวนข้อมูล	
		ขนาดข้อมูลจริง	ขนาดข้อมูลจริง
	16	16	16
	50	6	50
	256	8	256
	1,000	10	1,000
	10,000	11	10,000
	100,000	17	100,000
	1,000,000	20	1,000,000

การคำนวณค่าแฮชสำหรับข้อมูลจริงใช้การคำนวณค่าแฮชจริงจากข้อมูลจริง
ตามค่าจริง



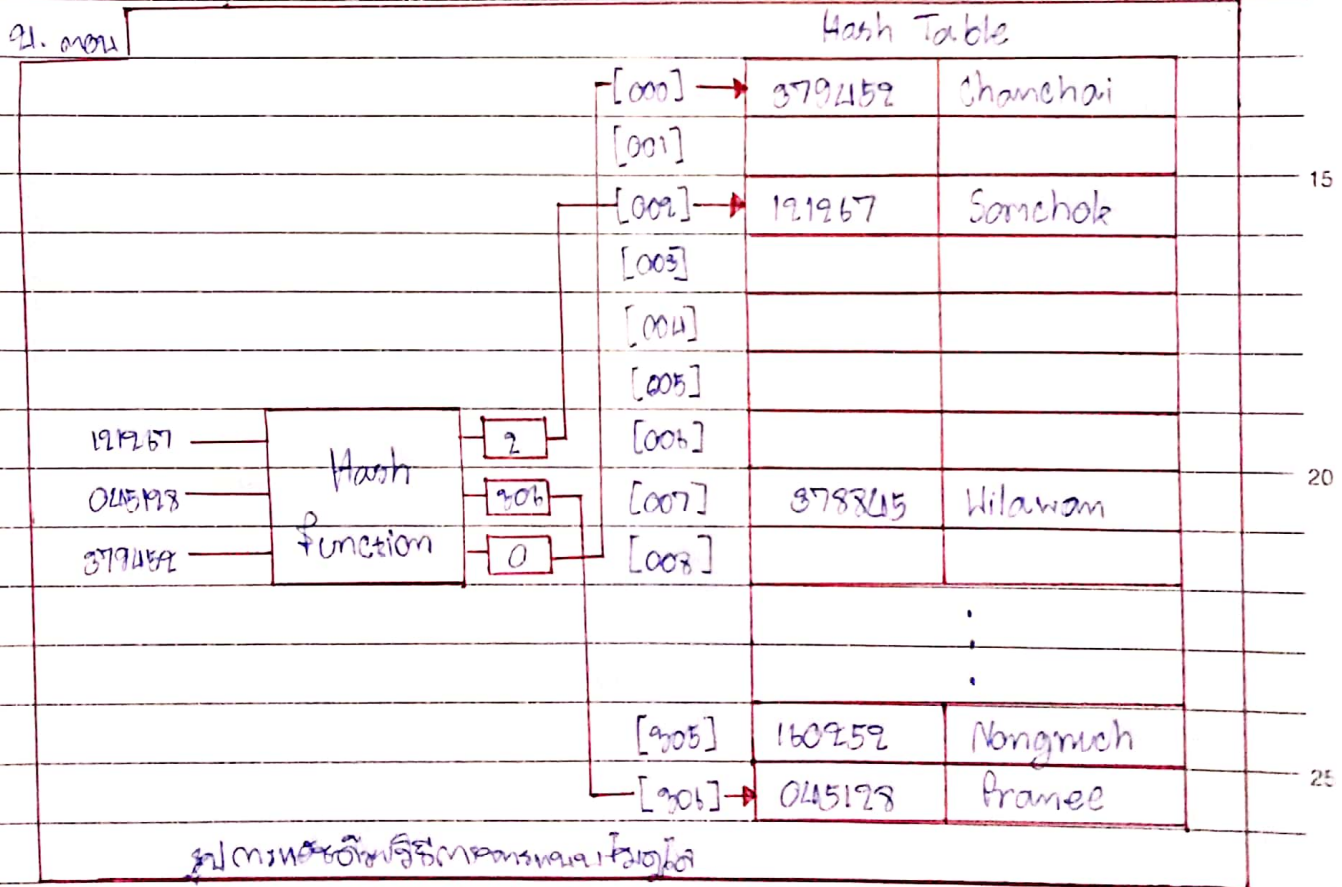
การคำนวณค่าแฮช

19. ๐๐๑ - คีย์ (Key) คือข้อมูลที่จะนำมาใช้เพื่อค้นหาข้อมูลที่ต้องการ
- ฟังก์ชันแฮช (Hash function) คือกระบวนการที่จะนำคีย์มาแปลงให้เป็นค่าแฮช
- ซึ่งค่าแฮชนั้นจะมีลักษณะเป็นตัวเลขที่มีค่าคงที่และไม่ขึ้นกับขนาดของคีย์
- ตารางแฮช (Hash Table) คือตารางที่ใช้เก็บค่าแฮชและข้อมูลที่ต้องการค้นหา

20. ๐๐๑ การคำนวณค่าแฮชของคีย์ "Somchok" โดยใช้ฟังก์ชันแฮชแบบง่าย ๆ สามารถทำได้ดังนี้

$$\text{Address} = \text{key} \text{ MODULO listSize}$$

โดยที่ listSize คือขนาดของตารางแฮช

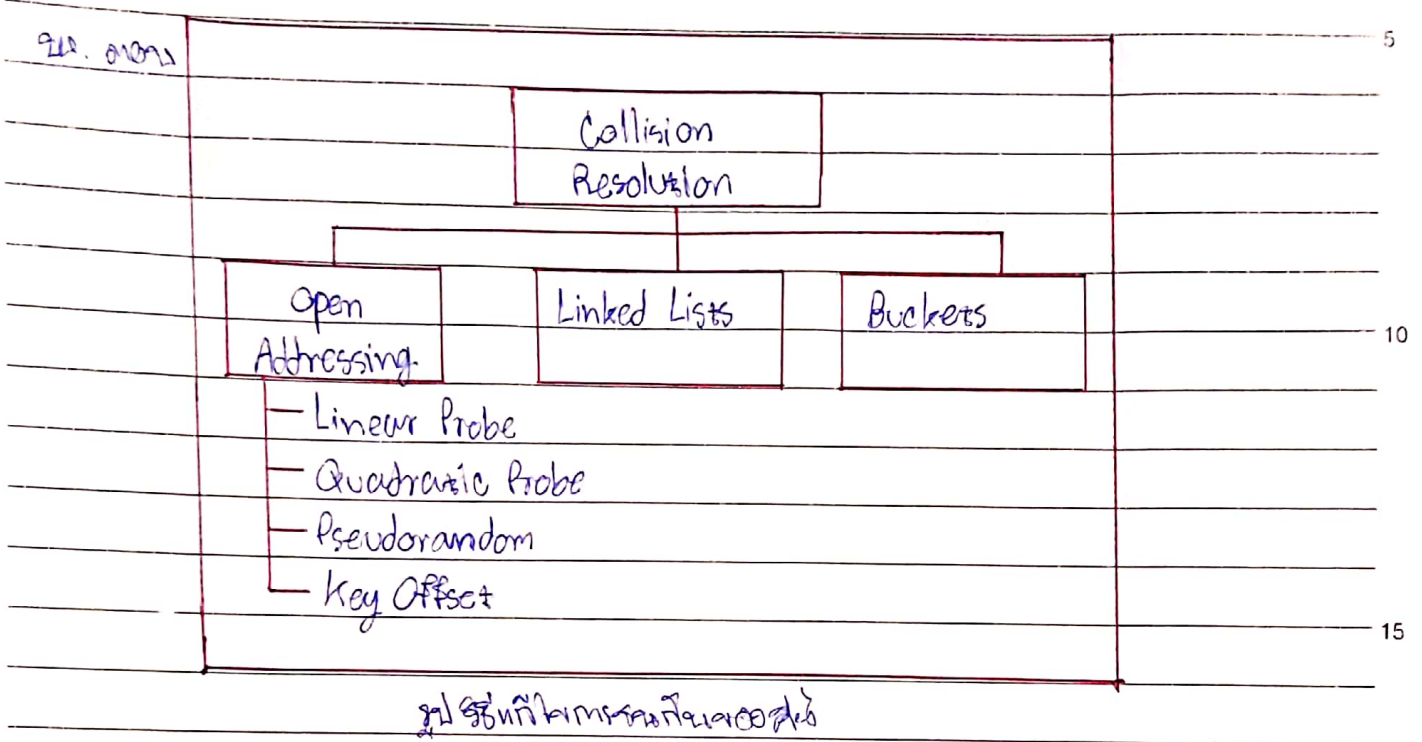


22. ๐๐๑ ตัวอย่าง การคำนวณคีย์ของข้อมูล "Somchok" โดยใช้คีย์ "121267" ที่กำหนด

$$121267 \% 101 = 9$$

$$\therefore \text{Hash}(121267) = 9$$

23. สมมุติว่าเรามีโปรแกรมที่เก็บข้อมูลแบบ Hash Table ซึ่งใช้วิธีการ Hashing เพื่อแปลงค่าของข้อมูลให้เป็นค่าที่อยู่ในช่วงของ Array ซึ่งค่าเหล่านี้จะถูกเก็บไว้ใน Array และเมื่อเราต้องการค้นหาข้อมูล เราก็ใช้ค่า Hash ที่ได้จาก Hash Function ไปค้นหาใน Array ซึ่งถ้าหากว่าค่า Hash ที่ได้จาก Hash Function ไม่ตรงกับค่าที่อยู่ใน Array เราก็จะทำการค้นหาซ้ำๆ ไปเรื่อยๆ จนกว่าจะเจอค่าที่ต้องการ หรือถ้าหากว่าค่า Hash ที่ได้จาก Hash Function ตรงกับค่าที่อยู่ใน Array เราก็จะทำการค้นหาซ้ำๆ ไปเรื่อยๆ จนกว่าจะเจอค่าที่ต้องการ



25. สมมุติว่า Linear Probe เป็นวิธีการแก้ปัญหา Collision โดยวิธีการนี้จะทำการค้นหาข้อมูลใน Array ที่ตำแหน่งที่ค่า Hash ตรงกับค่าที่อยู่ใน Array ถ้าหากว่าค่า Hash ที่ได้จาก Hash Function ไม่ตรงกับค่าที่อยู่ใน Array เราก็จะทำการค้นหาซ้ำๆ ไปเรื่อยๆ จนกว่าจะเจอค่าที่ต้องการ หรือถ้าหากว่าค่า Hash ที่ได้จาก Hash Function ตรงกับค่าที่อยู่ใน Array เราก็จะทำการค้นหาซ้ำๆ ไปเรื่อยๆ จนกว่าจะเจอค่าที่ต้องการ

26. สมมุติว่าวิธีการแก้ปัญหา Collision ด้วยวิธี Linear Probe จะมีข้อดีอยู่ 2 อย่างแรกอย่างแรกคือวิธีการนี้จะง่ายและสะดวกในการใช้งาน และอีกอย่างหนึ่งคือวิธีการนี้จะง่ายและสะดวกในการใช้งาน

27. สมมุติว่า Linear Probe เป็นวิธีการแก้ปัญหา Collision โดยวิธีการนี้จะทำการค้นหาข้อมูลใน Array ที่ตำแหน่งที่ค่า Hash ตรงกับค่าที่อยู่ใน Array ถ้าหากว่าค่า Hash ที่ได้จาก Hash Function ไม่ตรงกับค่าที่อยู่ใน Array เราก็จะทำการค้นหาซ้ำๆ ไปเรื่อยๆ จนกว่าจะเจอค่าที่ต้องการ หรือถ้าหากว่าค่า Hash ที่ได้จาก Hash Function ตรงกับค่าที่อยู่ใน Array เราก็จะทำการค้นหาซ้ำๆ ไปเรื่อยๆ จนกว่าจะเจอค่าที่ต้องการ

๒๘. ทราบ วิธี ลิงก์ ลิสต์ ในตารางอัตราการใช้ช่องฟรีส" เทคนิค (Chaining)" ซึ่งวิธีนี้ใช้วาง Link
 เชื่อมถึงได้เป็นแบบวงวนหรือแบบเส้นตรงก็ได้ และถ้าหากพบว่ามี Link ลิงก์ที่เชื่อมถึงกันแล้ว
 อีกวิธีหนึ่งคือวิธีการเก็บค่า Link ไปเก็บไว้ใน Array หรือ List ก็ได้เช่นกัน อีกวิธีหนึ่งคือวิธีการเก็บค่า Link
 ไปเก็บไว้ใน Map หรือ Dictionary ก็ได้เช่นกัน

๒๙. ทราบ วิธี ค้นหา ค่าเก็บข้อมูลในแบบ Hashing โดยทั่วไปแล้วการค้นหาข้อมูลใน Hashing นั้นจะใช้
 คำหลัก หรือ Keyword เป็นหลักในการค้นหา ซึ่งค่าหลัก (Prime Area) ซึ่งค่าหลักนี้จะเป็นค่าที่
 ใช้ในการค้นหา Home Address ใช้กันมาก

๓๐. ทราบ วิธี Bucket Hashing โดยวิธีนี้ใช้เก็บข้อมูลใน Bucket ซึ่งค่าเก็บข้อมูลนั้นจะเก็บ
 ไว้ใน Array หรือ List ก็ได้เช่นกัน และถ้าหากพบว่ามี Bucket ที่เก็บข้อมูลเหมือนกันแล้ว
 ก็ให้เก็บค่า Link ไปเก็บไว้ใน Array หรือ List ก็ได้เช่นกัน

สรุปท้ายบทที่ 10

การค้นหาลิสต์ คือ การหาตำแหน่งของข้อมูลที่ต้องการในลิสต์

เทคนิคการค้นหาลิสต์ ประกอบด้วย 3 วิธีพื้นฐาน คือ

1. การค้นหาแบบลิสต์

2. การค้นหาแบบไบนารี

3. การค้นหาแบบแฮช

การค้นหาลิสต์แบบลิสต์ คือ การค้นหาแบบลิสต์ที่ข้อมูลในลิสต์ไม่เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ต้องใช้หน่วยความจำเพิ่มขึ้น

การค้นหาลิสต์แบบไบนารี คือ การค้นหาแบบไบนารีที่ข้อมูลในลิสต์เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ข้อมูลในลิสต์ต้องเรียงลำดับกัน

การค้นหาลิสต์แบบแฮช คือ การค้นหาแบบแฮชที่ข้อมูลในลิสต์ไม่เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ต้องใช้หน่วยความจำเพิ่มขึ้น

การค้นหาลิสต์แบบลิสต์ คือ การค้นหาแบบลิสต์ที่ข้อมูลในลิสต์ไม่เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ต้องใช้หน่วยความจำเพิ่มขึ้น

การค้นหาลิสต์แบบไบนารี คือ การค้นหาแบบไบนารีที่ข้อมูลในลิสต์เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ข้อมูลในลิสต์ต้องเรียงลำดับกัน

ประสิทธิภาพของการค้นหาแบบลิสต์ คือ $O(n)$

ประสิทธิภาพของการค้นหาแบบไบนารี คือ $O(\log n)$

ประสิทธิภาพของการค้นหาแบบแฮช คือ $O(1)$

การค้นหาลิสต์แบบแฮช คือ การค้นหาแบบแฮชที่ข้อมูลในลิสต์ไม่เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ต้องใช้หน่วยความจำเพิ่มขึ้น

การค้นหาลิสต์แบบลิสต์ คือ การค้นหาแบบลิสต์ที่ข้อมูลในลิสต์ไม่เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ต้องใช้หน่วยความจำเพิ่มขึ้น

การค้นหาลิสต์แบบไบนารี คือ การค้นหาแบบไบนารีที่ข้อมูลในลิสต์เรียงลำดับกัน โดยวิธีนี้จะเพิ่มประสิทธิภาพในการค้นหาข้อมูลที่ต้องการได้เร็วกว่าวิธีอื่น ๆ แต่ข้อมูลในลิสต์ต้องเรียงลำดับกัน