

Hash table concept (linklist)

* ใช้มันทำโครงสร้างข้อมูลแบบ plain text
* ใช้มันจัดเก็บ Hash table (C++ lib)

hash = ช่วยหาข้อมูลอย่างรวดเร็ว, ใช้เวลาใน 1 วินาทีคือถือว่าเร็ว 100,000 ครั้งต่อวินาที

→ ออกแบบโครงสร้างข้อมูลให้ $\rightarrow O(1)$

keys $\xrightarrow{\text{hash function}}$ bucket

hash function ที่ง่ายที่สุด $f(x) = x$ ไม่สนใจ, แต่ถ้าข้อมูลมี 4,000,000,000 ตัว จะ insert 50,000,000 ตัวจะช้ามาก

Array

Linklist

find, memory usage, $O(n)$ ก็ยังโอเคอยู่

hash function \rightarrow map เลขไปเรียงๆ, เปลี่ยน object \rightarrow index

→ เรียง

→ จัดทำตารางที่เก็บข้อมูลไว้เรียงกัน

→ ตารางใน 1 ตาราง hash แล้วจะหาเจอ hash ที่ไปเจอที่นี้

ถ้า hash ค่าจะชนกันได้เหมือนกัน = collision ไม่ค่อยเกิดขึ้น

9-2

→ อยากรู้ Collision อยากรู้

Map, Dict, Associative array ออกสอบง่ายไม่ยาก

→ ถ้า key ไม่ = value id (ชื่อ, รหัสประจำตัว), (เลขของเบอร์, เบอร์โทร)

Hash key ออกสอบ

Set (o) ออกสอบ

Runtime Analysis

ถ้า c เป็นค่าบวกที่คงที่ใน $\rightarrow O(c+1)$

Memory $O(n-m)$

Set \rightarrow ทำหน้าที่เก็บข้อมูลว่าไม่ซ้ำกัน

Implement Set \rightarrow

Add (o) \rightarrow S.add

$L \leftarrow A[L(o)]$

for o' in L:

if o' == o

return

L.append

สรุป

\rightarrow object \rightarrow hash = chaining

\rightarrow Mem = $O(n+m)$

\rightarrow operation = $O(c+1)$

ไม่สนใจอะไรเลย จบ สัปดาห์ 1 จบแล้ว

Hash function / Integer, parse Int (s) ออกสอบ.

int (123-45-67) = 1,234,567

10^L \Rightarrow for real?!!

ถ้าได้ 10¹ / 10² ขึ้นไป

for chaining

ก่อนนี้มันจะจัดเก็บ

parameter $n \leftarrow \text{num}$, $m \leftarrow \text{cardinal}$, $c \leftarrow \text{constant}$

load factor $\rightarrow \alpha = \frac{n}{m}$ ออกสอบด้วย

First Digits

\rightarrow ไม่สนใจ Area code

Last Digits

\rightarrow ถ้าเจอเลขที่ 5 = 10

Random

\rightarrow การกระจายของตัวเลข, ทำแบบสุ่มๆ

Hash function ที่ 10

\rightarrow จำนวนที่ได้ (ตัวเลข)

1, 2

\rightarrow 6 หลักด้วย

ไม่มี hash function ก็ยังใช้ได้ เพราะมันยังใช้กับมัน

\rightarrow 1000000 คืออัตรา $\frac{1}{m}$

Universal form of hash \leftarrow ฟังก์ชันที่ 10

อัตรา $m \leq \frac{1}{\alpha}$ ถ้า α มีค่า

runtime $O(1+\alpha^m)$, $\alpha = \frac{n}{m}$ หรือ bad factor ที่มันใช้

มีอัตรา m

= load factor ที่ 10 $0.5 < \alpha < 1.0$ \leftarrow ค่าที่ 10

runtime $\rightarrow O(1)$ ถ้า α มีค่า

ถ้า α มีค่า 10, 100, 1000

Rehash (t) ออกสอบแบบไม่ยาก

if $\alpha > 0.9 \rightarrow$ สร้าง 2x Size

Hash function

\rightarrow ถ้าเป็น int \rightarrow เปลี่ยนเป็น int \rightarrow 2x Size

parameter ที่ 10 \rightarrow Lemma 2 =

$(ax+b) \% p$ \rightarrow ฟังก์ชันที่ 10 (Linear hash) ออกสอบง่าย

String hashing algo

| | = ขนาดของ String

ถ้า s = string

$s = s[0]s[1]s[2]$

ถ้า int $a = (int)s.charAt[0]$ ถ้า $a = 0$ แล้ว

Polynomial Hashing

$\sum_{i=0}^{L-1} s[i]x^i \% p$ \rightarrow ฟังก์ชันที่ 10 (หรือ 9-1)

อัตรา $m \leq \frac{1}{\alpha}$ ถ้า α มีค่า

Runtime $c = O(1+\alpha)$

สรุป \rightarrow Hash มีหลาย Hash String

10 หลัก 2 หลัก