Lec08 全域哈希与完全哈希

addressing a fundamental weakness of hashing,

for any choice of hash function, there exists a bad set of keys that all hash to the same slot,

the idea is to choose a hash function at random, independently from the keys

the name of the scheme is universal hashing (全域哈希)

Def. let U be a universe of keys, and let H be a finite collection of hash functions, mapping U to the slots in our hash table $\{0, 1, \cdots, m-1\}$, say that H is universal if for all pairs of distinct keys ($\forall x, y \in U$ and $x \neq y$), the following is true:

$$\left|\{h \in H : h(x) = h(y)\}\right| = \frac{|H|}{m}$$

"在函数集H中，对于任意键对，能将它们(指键对)哈希映射到同个位置的哈希函数的数目等于 $\frac{|H|}{m}$"

"也可以这样看，如果哈希函数h是随机地从函数集H里选出的，那么x与y发生碰撞的概率是 $\frac{1}{m}$"

Thm. choose h randomly from H, suppose we're hashing n keys into m slots in table T, then for given key x, the expected number of collisions with x is less than $\frac{n}{m}$, i.e. $E(\#\text{collisions with } x) < \frac{n}{m}$.

"α: the load factor of the table"

proof:

let C_x be the random variable denoting the total number of collisions of keys in T with x, and let $c_{xy} = \begin{cases} 1, & \text{if } h(x) = h(y) \\ 0, & \text{otherwise} \end{cases}$

note that $E(c_{xy}) = \frac{1}{m}$ and $C_x = \sum_{y \in T, y \neq x} c_{xy}$

$$\begin{aligned} E(C_x) &= E\Big(\sum_{y \in T, y \neq x} c_{xy}\Big) \\ &= \sum_{y \in (T - \{x\})} E(c_{xy}) \quad \longleftarrow \text{期望的线性性质} \\ &= \sum_{y \in (T - \{x\})} \frac{1}{m} \\ &= \frac{n-1}{m} \quad \text{Q.E.D.} \end{aligned}$$

constructing an universal hash function (一种构造全域哈希的方法)

① let m be prime (质数),
decompose any key k in our universe into $r+1$ digits: $k = \langle k_0, k_1, \cdots, k_r \rangle$, $0 \leq k_i \leq m-1$
"这种做法的思想是把k用m进制来表示"

② pick an a at random, $a = \langle a_0, a_1, \cdots, a_r \rangle$ "同样的，我们也把a看成是m进制数",
each a_i is chosen randomly from $\{0, 1, 2, \cdots, m-1\}$, "a_i是随机的m进制数,

③ define $h_a(k) = \Big(\underbrace{\sum_{i=0}^{r} a_i k_i}_{\vec{a}\text{与}\vec{k}\text{的点乘}}\Big) \bmod m$

how big is $H = \{h_a\}$? ans: m^{r+1}

Thm: H is universal.

proof: let $x = <x_0, x_1, \cdots, x_r>$

$y = <y_0, y_1, \cdots, y_r>$ be distinct keys

x and y differ in at least one digit, without loss of generality, position 0.

for how many hash functions $h_a \in H$, do x and y collide?

must have $h_a(x) = h_a(y)$ if they collide

$$\Rightarrow \sum_{i=0}^{r} a_i x_i \equiv \sum_{i=0}^{r} a_i y_i \pmod m$$

$$\Rightarrow \sum_{i=0}^{r} a_i (x_i - y_i) \equiv 0 \pmod m$$

$$\Rightarrow a_0(x_0 - y_0) + \sum_{i=1}^{r} a_i (x_i - y_i) \equiv 0 \pmod m$$

$$\Rightarrow a_0(x_0 - y_0) \equiv -\sum_{i=1}^{r} a_i (x_i - y_i) \pmod m$$

number theory fact: let m be prime, for any $z \in \mathbb{Z}_m$ (integers mod m), $z \neq 0$, $\exists$ unique $z^{-1} \in \mathbb{Z}_m$, $z \cdot z^{-1} \equiv 1 \pmod m$

Example:

$m = 7$

z	1	2	3	4	5	6
z^{-1}	1	4	5	2	3	6

since $x_0 \neq y_0$, $\exists (x_0 - y_0)^{-1}$,

$$\Rightarrow a_0 \equiv \left(-\sum_{i=1}^{r} a_i (x_i - y_i)\right) \cdot (x_0 - y_0)^{-1}$$

"$a_0, a_1, a_2, \cdots, a_r$ 线性相关"

★ 若两个互异的键被哈希到同一个位置上，那么a_0实际上由其它所有的a_i所决定

thus, for any choice of $a_1, a_2, \cdots, a_r$, exactly 1 of the m choices for a_0 will cause x and y to collide, and no collision for other m-1 choices for a_0

so the number of hash functions that cause x and y to collide

$$= \underbrace{m}_{\text{choices for } a_1} \cdot \underbrace{m}_{\text{choices for } a_2} \cdot \cdots \cdot \underbrace{m}_{\text{choices for } a_r} \cdot \underbrace{1}_{\text{choices for } a_0}$$

choices for a_0, this value $\left(-\sum_{i=1}^{r} a_i(x_i - y_i)\right) \cdot (x_0 - y_0)^{-1}$

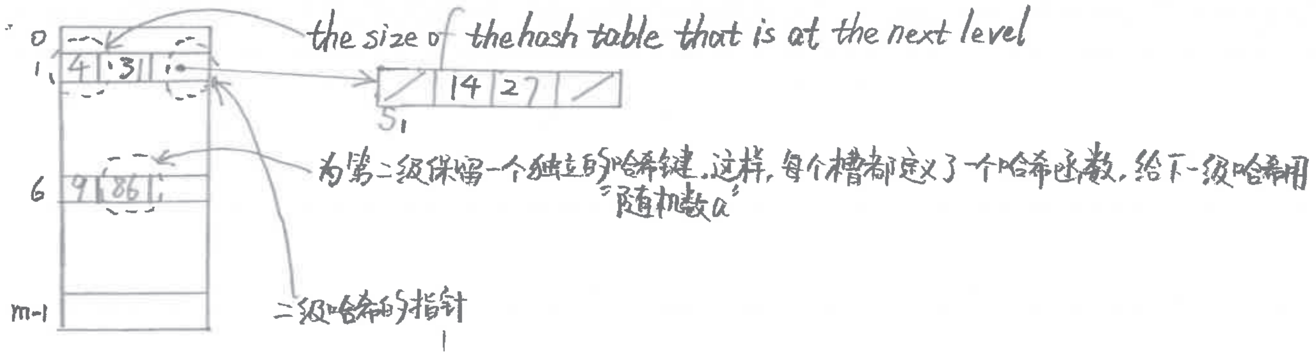
$$= m^r = \frac{|H|}{m} \quad \text{Q.E.D.}$$

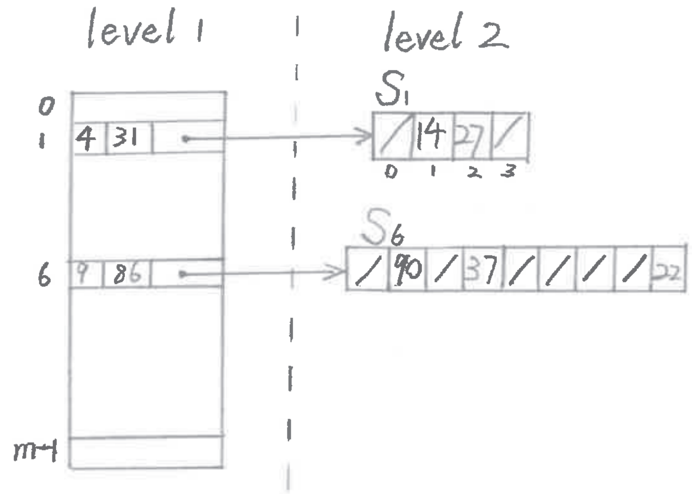
perfect hashing (完全哈希)

"suppose I give you a set of keys, build a static table for me, so I can look up whether the key is in the table"

given n keys, construct a static hash table of size $m = O(n)$, such that search takes $O(1)$ time in the worst case

the idea is to use a two-level scheme (双级结构), with universal hashing at both levels, so that no collisions at level two





$h(14)=h(27)=1$ $\quad$ $h_{31}(14)=1$

$h_{31}(27)=2$

如果能保证在第二级没有碰撞,那么只需要花费 O(1) 的时间就能在最坏情况下完成对数据的查找。

if n_i items that hash to level-one's slot i, then use $m_i = n_i^2$ slots in the level-two hash table.

"此时,第二级表将会非常稀疏"

and what I am going to show is that under those circumstances, it's easy for me to find hash functions such that there are no collisions.

Analysis for Level 2

Thm. hash n keys into $m=n^2$ slots, using a random hash function in an universal set H, then the expected number of collisions is less than $\frac{1}{2}$.

proof: the probability that 2 given keys collide under h is $\frac{1}{m} = \frac{1}{n^2}$,

C_n^2 pairs of keys.

therefore, $E(\#\text{collisions}) = C_n^2 \cdot \frac{1}{n^2} = \frac{1}{2} \cdot \frac{n-1}{n} < \frac{1}{2}$ Q.E.D.

Markov Inequality

for random variable x which is bounded below by 0,

$P\{x \geq t\} \leq \frac{E(x)}{t}$

proof:

$$E(x) = \sum_{x=0}^{\infty} x \cdot P(x)$$
$$\geq \sum_{x=t}^{\infty} x \cdot P(x)$$
$$\geq \sum_{x=t}^{\infty} t \cdot P(x)$$
$$= t \cdot \sum_{x=t}^{\infty} P(x)$$
$$= t \cdot P(x \geq t) \quad Q.E.D.$$

Corollary: $P\{\text{no collisions}\} \geq \frac{1}{2}$

proof: $P\{\text{at least one collision}\} \leq E(\#\text{collisions})/1$

$$< \frac{1}{2}$$

$P\{0 \text{ collision}\} = 1 - P\{\geq 1 \text{ collision(s)}\} \geq \frac{1}{2}$ Q.E.D.

So to find a good level-2 hash function,
just test a few at random, and we will find one quickly,
since at least half will work. (可行性分析)

Analysis for Storage (证明是 $O(n)$ 大小)

for level 1, choose $m = n$,
and let n_i be the random variable for the number of keys that hash to slot i in T
use $m_i = n^2$ slots in each level 2 table S_i

$E(\text{total storage}) = n + E\left(\sum_{i=0}^{m-1} \Theta(n_i^2)\right)$ ⟵ 桶排序里的知识

$= \Theta(n)$ by bucket-sort analysis