9.词典

（d2） 散列：排解冲突(2)

我真的以为

这样何尝不是一种所谓的解脱

要背负的辛苦又有谁能够清楚

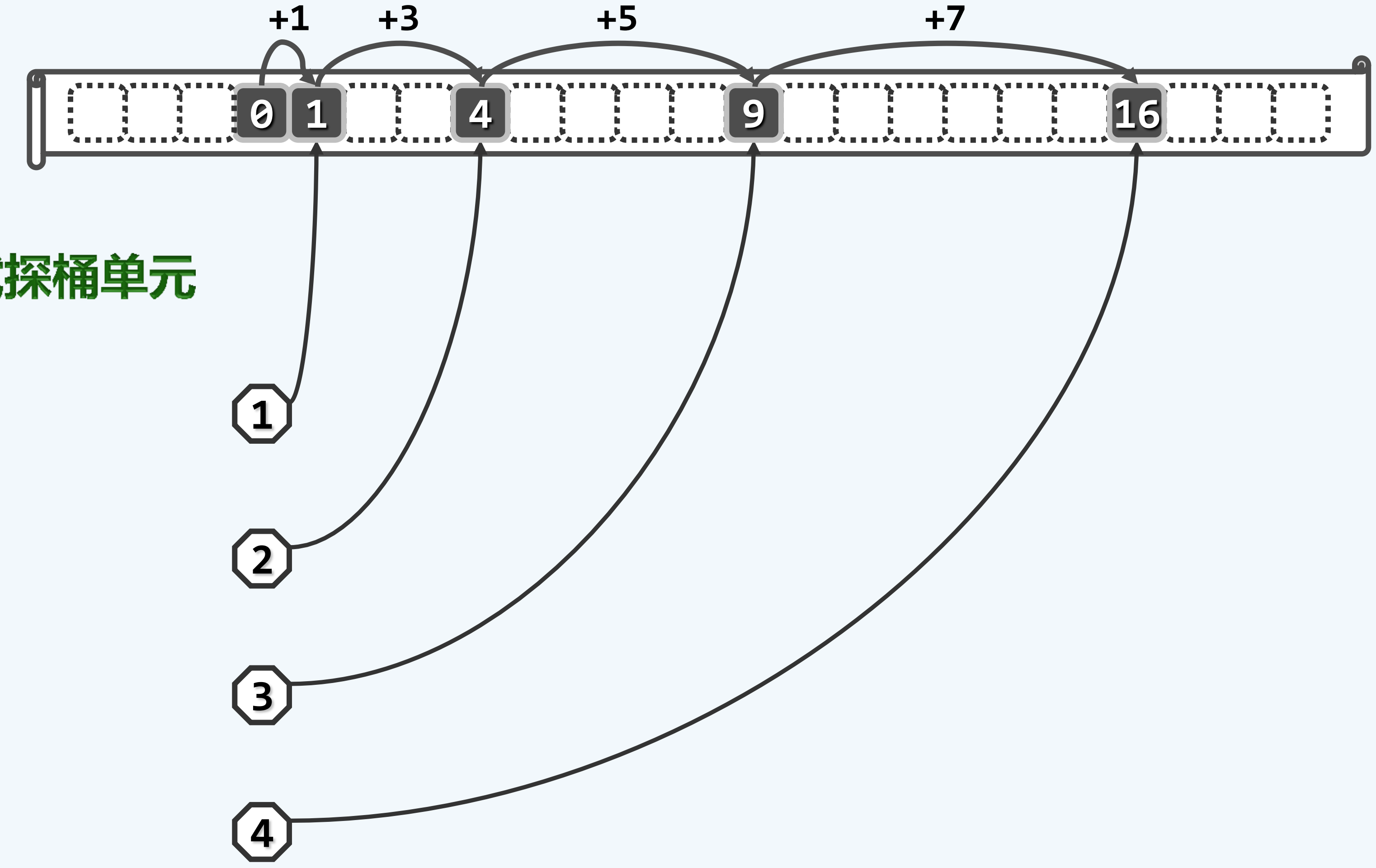
那内心的冲突

邓俊辉

deng@tsinghua.edu.cn

平方试探

❖ Quadratic probing



以平方数为距离，确定下一试探桶单元

[hash(key) + $\boxed{1^2}$] % M

[hash(key) + $\boxed{2^2}$] % M

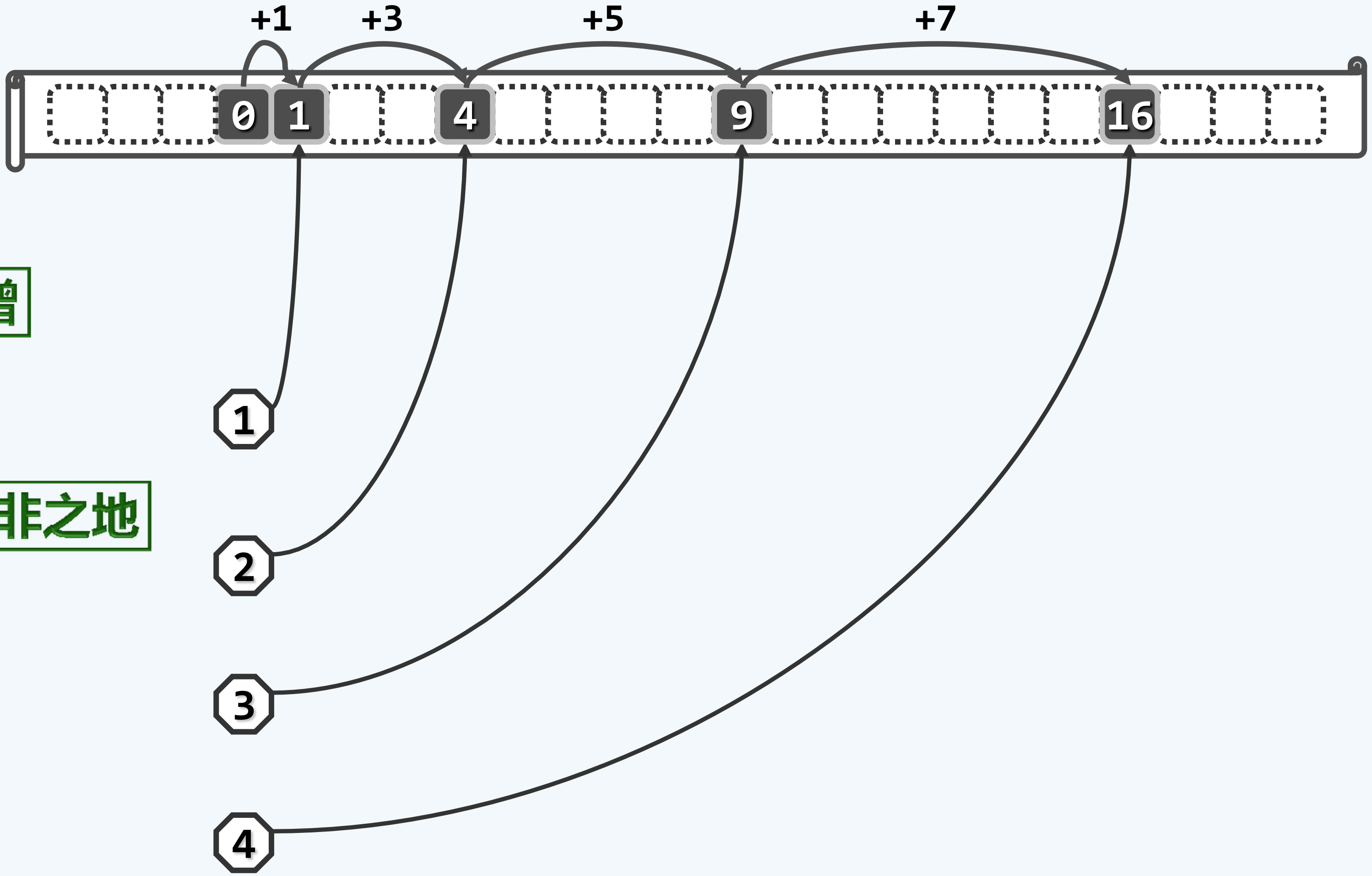
[hash(key) + $\boxed{3^2}$] % M

[hash(key) + $\boxed{4^2}$] % M

...

优点、缺点及疑惑

❖ 数据聚集现象有所缓解



查找链上，各桶间距线性递增

一旦冲突，可聪明地跳离是非之地

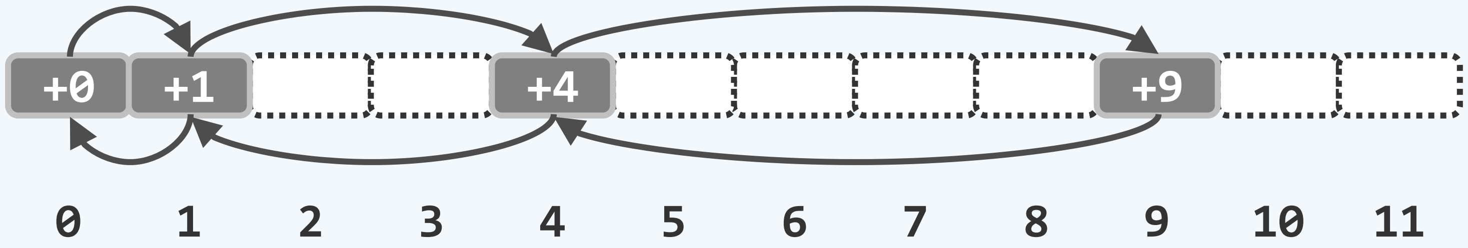
❖ 若涉及外存，I/O将激增

❖ 只要有空桶，就...一定能...找出来吗？

//毕竟不是挨个试探

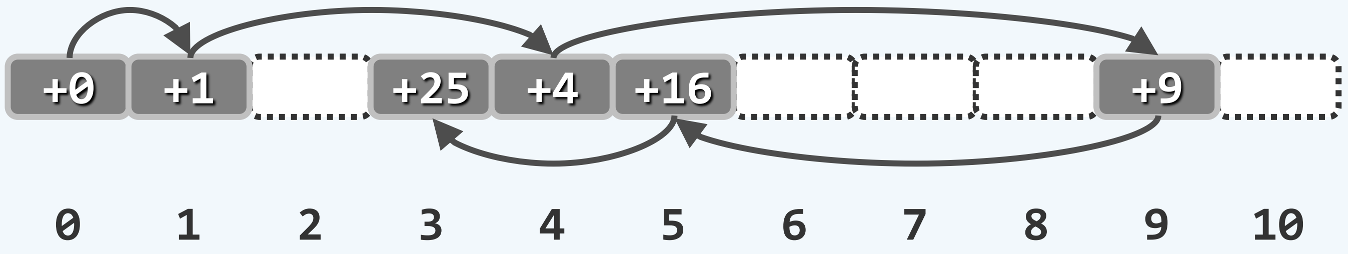
装填因子，须足够小！

❖ $\{\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ \ldots\ \}^2\ \%\ \boxed{12}\ =\ \{\ 0,\ 1,\ 4,\ 9\ \}$



M若为合数：$n^2\ \%\ M$可能的取值必然少于$\lceil M/2 \rceil$种——此时，只要对应的桶均非空...

❖ $\{\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ \ldots\ \}^2\ \%\ \boxed{11}\ =\ \{\ 0,\ 1,\ 4,\ 9,\ 5,\ 3\ \}$



M若为素数：$n^2\ \%\ M$可能的取值恰好会有$\lceil M/2 \rceil$种——此时，恰由查找链的前$\lceil M/2 \rceil$项取遍

❖ 定理：若M是素数，且$\lambda \le 0.5$，就一定能够找出；否则，不见得

查找链前缀，必足够长！

❖ 反证：假设存在 $\boxed{0 \le a < b < \lceil M/2 \rceil}$，使得

沿着查找链，$\boxed{\text{第}a\text{项}}$和$\boxed{\text{第}b\text{项}}$彼此冲突

❖ 于是：a^2和b^2自然属于M的某一$\boxed{\text{同余类}}$，亦即

$$a^2 \equiv b^2 \pmod M$$

$$b^2 - a^2 = (b + a)\cdot(b - a) \equiv 0 \pmod M$$

❖ 然而：$\boxed{0 < \boxed{b - a} < \boxed{b + a} < M}$ —— 这与$\boxed{\text{M为素数}}$矛盾

❖ 那么，$\boxed{\text{另一半}}$的桶，可否也利用起来呢...

双向平方试探

❖ 自冲突位置起，依次向后试探

[hash(key) $\boxed{+\ 1^2}$] % M

[hash(key) $\boxed{-\ 1^2}$] % M

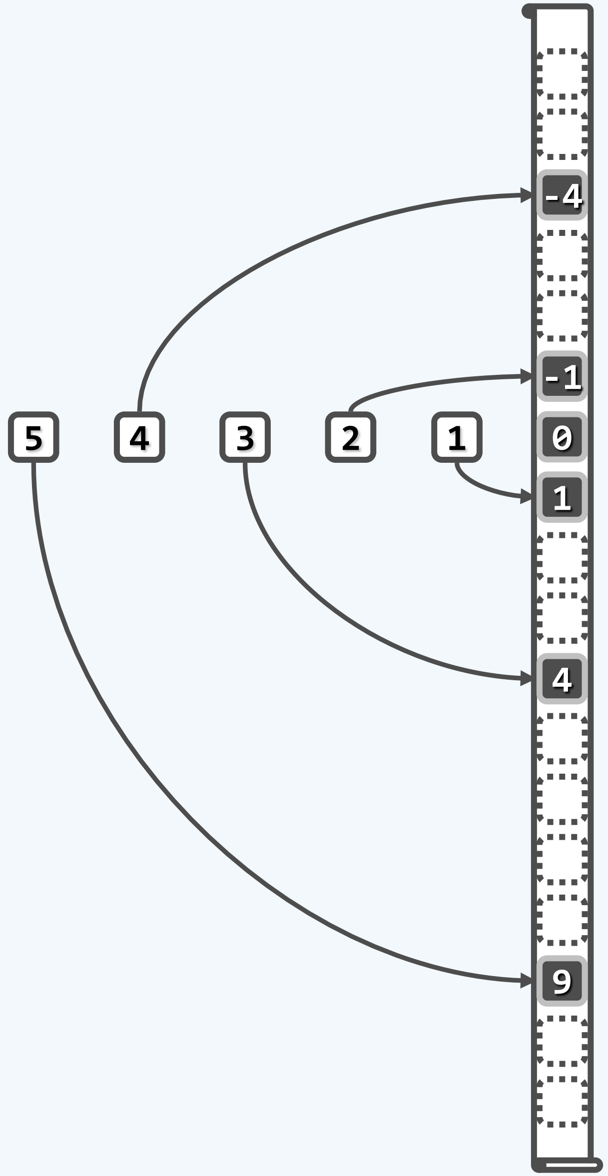
[hash(key) $\boxed{+\ 2^2}$] % M

[hash(key) $\boxed{-\ 2^2}$] % M

[hash(key) $\boxed{+\ 3^2}$] % M

[hash(key) $\boxed{-\ 3^2}$] % M

...



查找链，彼此独立？

❖ 正向和逆向的子查找链，各包含 $\lceil M/2 \rceil$ 个互异的桶

$-\lfloor M/2 \rfloor$, ... , -2 , -1 , $\boxed{0}$, 1 , 2 , ... , $\lfloor M/2 \rfloor$

± i^2		-36	-25	-16	-9	-4	-1	0	1	4	9	16	25	36
M	5					1	4	0	1	4				
	7				5	3	6	0	1	4	2			
	11		8	6	2	7	10	0	1	4	9	5	3	
	13	3	1	10	4	9	12	0	1	4	9	3	12	10

❖ 除了 0，这两个序列是否还有...其它公共的桶？

4k + 3

- 两类素数： $\boxed{3}$ 5 $\boxed{7}$ $\boxed{11}$ 13 17 $\boxed{19}$ $\boxed{23}$ 29 $\boxed{31}$ ...
- 表长取作素数 $\boxed{M = 4 \times k + 3}$ ，$\boxed{\text{必然可以}}$ 保证查找链的前 $\boxed{M}$ 项均互异

± i^2		-36	-25	-16	-9	-4	-1	0	1	4	9	16	25	36
M	5					1	4	0	1	4				
	7				5	3	6	0	1	4	2			
	11		8	6	2	7	10	0	1	4	9	5	3	
	13	3	1	10	4	9	12	0	1	4	9	3	12	10

- 反之， $\boxed{M = 4 \times k + 1}$ 就... $\boxed{\text{必然不可}}$ 使用？

双平方定理

❖ Two-Square Theorem of Fermat

任一素数 p 可表示为一对整数的平方和，当且仅当

$$\boxed{p \% 4 = 1}$$

❖ 只要注意到：

$$(u^2 + v^2)\cdot(s^2 + t^2) = (us + vt)^2 + (ut - vs)^2$$

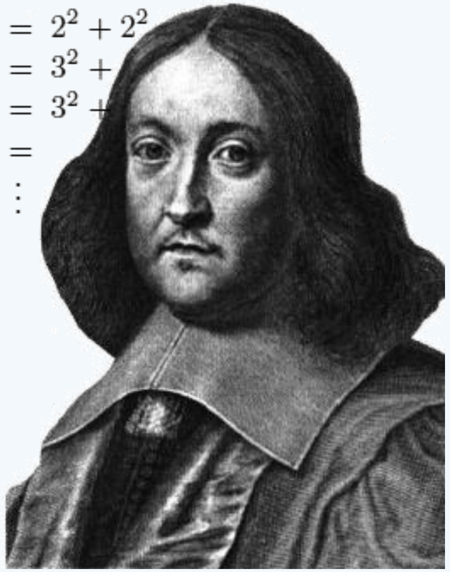
$$(2^2 + 3^2)\cdot(5^2 + 8^2) = (10 + 24)^2 + (16 - 15)^2$$

❖ 就不难推知：

任一自然数 n 可表示为一对整数的平方和，当且仅当

在其素分解中，　形如 $\boxed{M = 4 \times k + 3}$ 的每一素因子均为 偶数 次方

$$\begin{aligned}
1 &= 1^2 + 0^2 \\
2 &= 1^2 + 1^2 \\
3 &= \\
4 &= 2^2 + 0^2 \\
5 &= 2^2 + 1^2 \\
6 &= \\
7 &= \\
8 &= 2^2 + 2^2 \\
9 &= 3^2 + \\
10 &= 3^2 + \\
11 &= \\
&\vdots
\end{aligned}$$



（伪）随机试探法

❖ 原理： 自冲突位置起，“随机”试探下一位置

若冲突，则继续试探

❖ 问题： 这样...可行吗？

进行操作前，如何才能找到目标词条呢？ //请自己说服自己

❖ 同样，需要留意此方法跨平台的 兼容性 和 可移植性

再散列

❖ double hashing

第二散列函数：hash2()

发生冲突后，以hash2(key)为偏移量增量，重新确定地址

[hash(key) + $\boxed{1}$ × hash2(key)] % M

[hash(key) + $\boxed{2}$ × hash2(key)] % M

[hash(key) + $\boxed{3}$ × hash2(key)] % M

...直到发现一个空桶

❖ hash2()为常值函数时，即退化为...

重散列

```
❖ template <typename K, typename V> //装填因子增大，将导致冲突激增
  void Hashtable<K, V>::rehash() { //必要时，需“集体搬家”至更大的表
     int old_capacity = M; Entry<K, V>** old_ht = ht; N = 0;
     ht = new Entry<K, V>*[ M = primeNLT( 2*M ) ]; //新表容量至少加倍
     memset( ht, 0, sizeof( Entry<K, V>* ) * M ); //初始化各桶
     release(lazyRemoval); lazyRemoval = new Bitmap(M); //新建位图，容量至少加倍
     for ( int i = 0; i < old_capacity; i++ ) //扫描原桶数组
        if ( old_ht[i] ) //将非空桶中的词条逐一
           put( old_ht[i]->key, old_ht[i]->value ); //插入至新表
     release( old_ht ); //释放原桶数组
  }
```