

词典

散列：原理

09-A2

书者，散也。欲书先散怀抱，任情恣性，然后书之。

“如果是活着的人，想谁，找谁一趟不就完了？”

“找不得，找不得，当年就是因为个找，我差点丢了命。”

邓俊辉

deng@tsinghua.edu.cn

电话: 号码 ~ 人

### General inquiries

Tel: Toll Free: 1-800-IBM-4YOU

E-mail: [askibm@vnet.ibm.com](mailto:askibm@vnet.ibm.com)

[www.ibm.com/us/en/](http://www.ibm.com/us/en/)

### Shopping

Tel: Toll Free: 1-888-SHOP-IBM

### Sales Center

1-855-2-LENOVO (1-855-253-6686)

Mon - Fri: 9am-9pm (EST)

Sat - Sun: 9am-6pm (EST)

### Customer Service

1-855-2-LENOVO (1-855-253-6686)

Mon - Fri: 9am-9pm (EST)

Sat - Sun: 9am-6pm (EST)



# 电话簿

❖ 需求：为一所学校制作电话簿

号码 --> 教员、学生、员工、办公室

❖ 蛮力：用户记录 ~ 数组Rank ~ 电话号码

$O(1)$ 效率!

❖ 以清华为例 (2003)

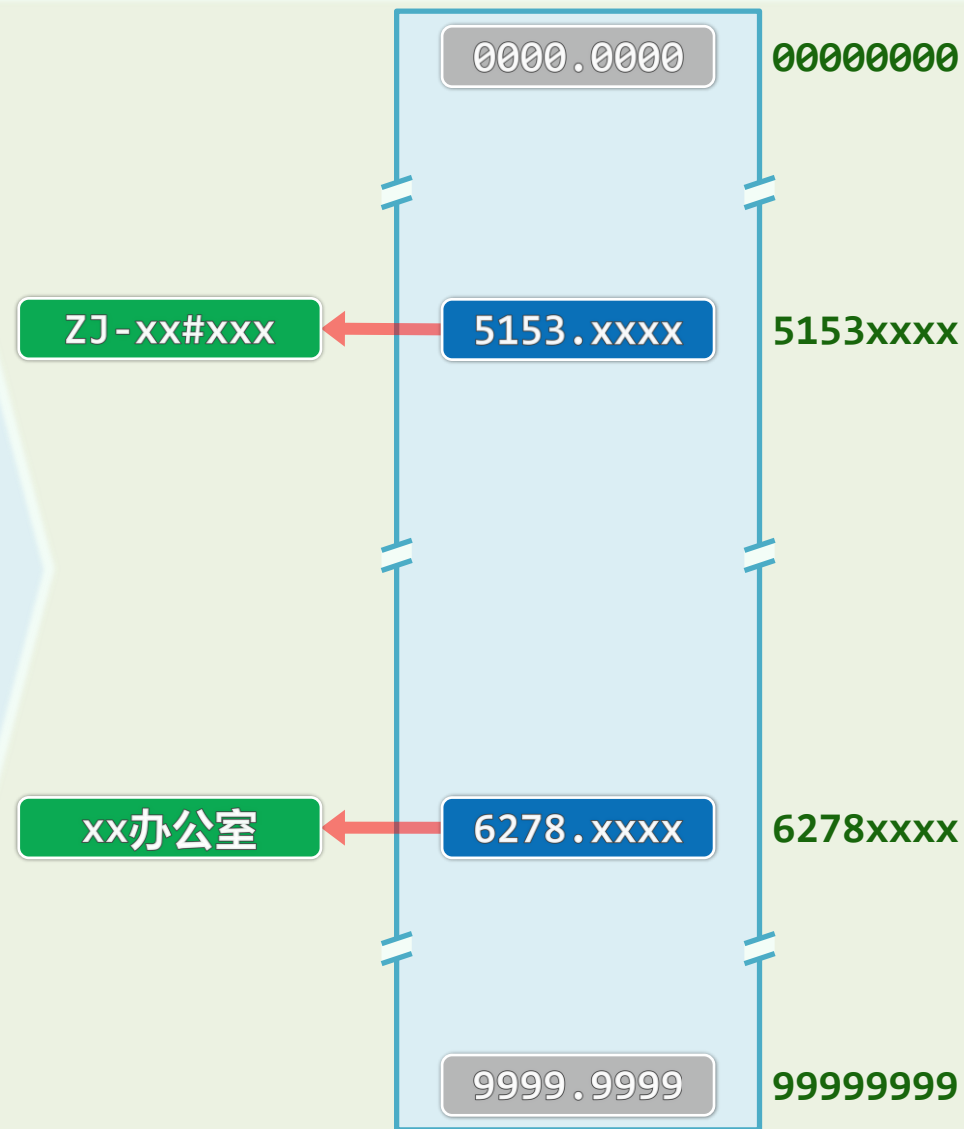
#可能的电话 =  $\mathcal{R}$  =  $10^8$  = 100M

#实有的电话 =  $\mathcal{N}$  = 25,000 = 25K

❖ 问题：空间 =  $O(\mathcal{R} + \mathcal{N})$  =  $O(100M + 25K)$

效率 =  $25K / 100M = 0.025\%$

❖ 如何在保持查找速度的同时，降低存储消耗?



# 散列表 / 散列函数

❖ 桶 (bucket) : 直接存放或间接指向一个词条

❖ Bucket array ~ Hashtable

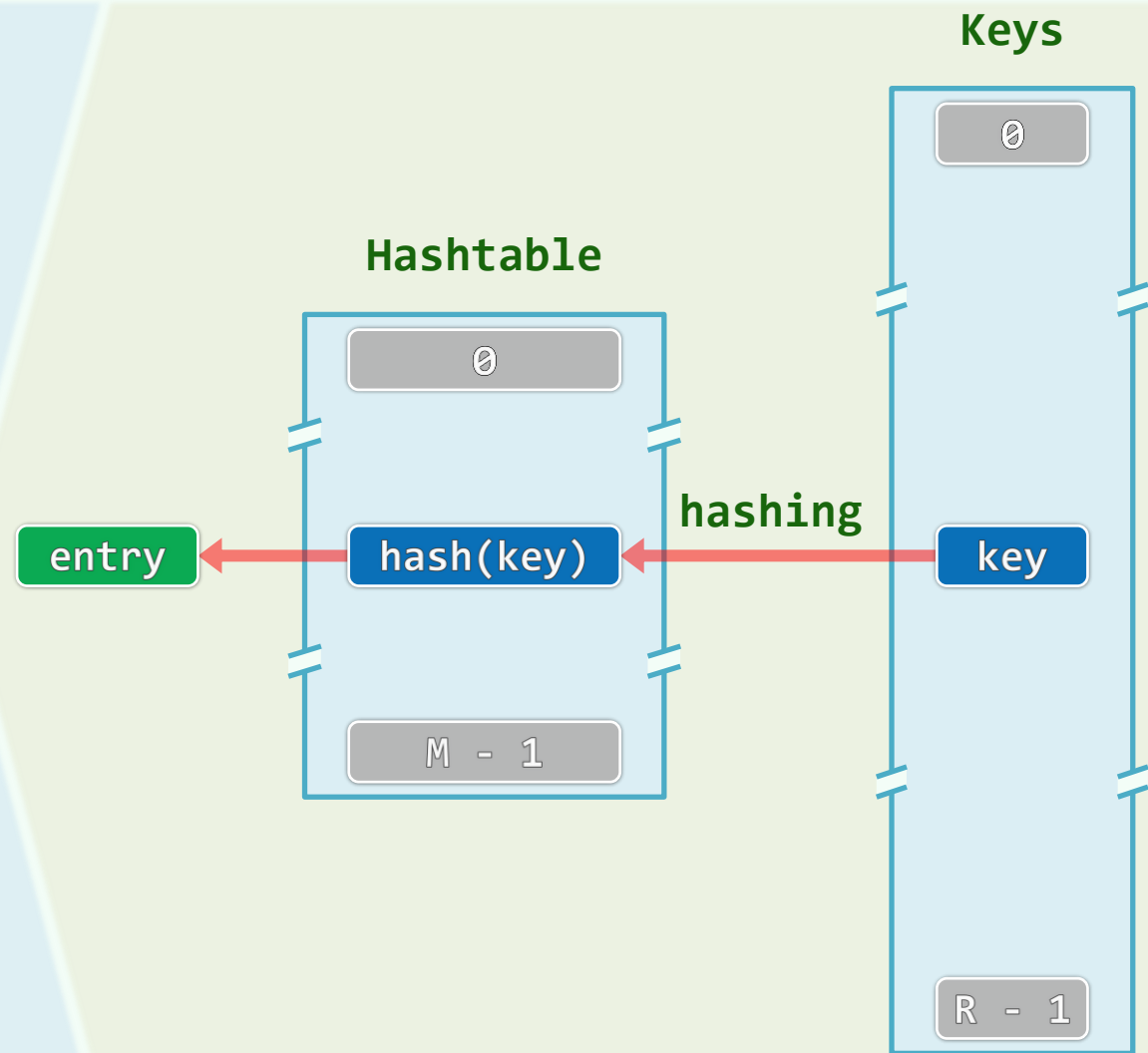
- 容量:  $\mathcal{M}$
- 满足:  $\mathcal{N} < \mathcal{M} \ll \mathcal{R}$
- 空间:  $\mathcal{O}(\mathcal{N} + \mathcal{M}) = \mathcal{O}(\mathcal{N})$

❖ 定址/杂凑/散列

- 根据词条的key (未必可比较)
- “直接” 确定散列表入口 (无论表有多长)

❖ 散列函数:  $hash() : key \mapsto \&entry$

❖ “直接” :  $expected - \mathcal{O}(1) \neq \mathcal{O}(1)$



# 实例

