# 计数排序

count=[0]\*100

self.arr

0数值出现了多少次

1数值出现了多少次

arr=self.arr

count[arr[i]]+=1

Arr

000000000 11111111

K

# 练习

O(nlogk)

堆排

前K个元素建立小根堆，遍历数组元素，发现比堆顶元素要大，淘汰堆顶元素，把这个元素放入堆顶，重新调整为小根堆，循环往复，到最后

N

右边

N/2

N/4

N/8

O(N)

1 2 3 4 5 6 7 8 9 10

# 位图

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 0000 0000

512M

# 哈希表

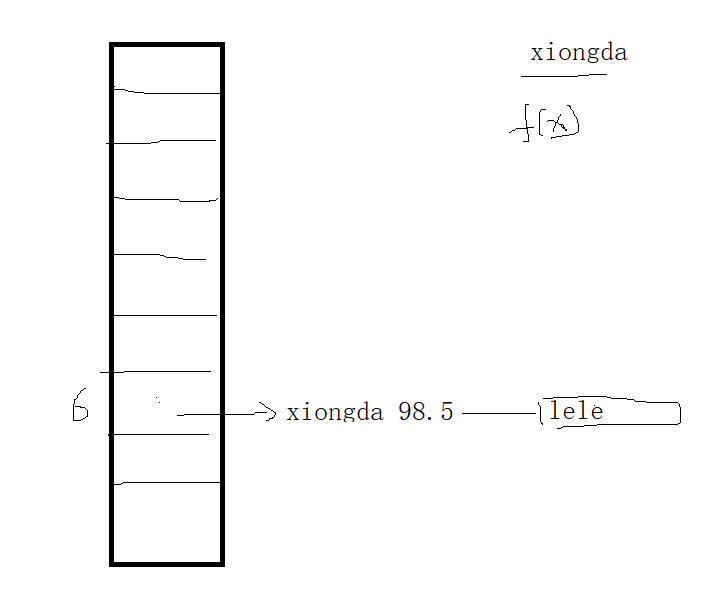
用到一个列表，也是我们的哈希表，大小是1000

elf\_hash(x) 得到的下标值，必须是0-999之间

不同字符串的elf\_hash(x)计算得到的下标值相同，那么就出现了哈希冲突

海量字符串 200G

文件名 7952993560868598381



# 字典树

用于字符串的去重

# 红黑树

[87, 2, 70, 41, 5, 55, 1, 67, 48, 86]

<https://www.cs.usfca.edu/~galles/visualization/Algorithms.html?tdsourcetag=s_pcqq_aiomsg>