# 9.词典

## （e）桶/计数排序

邓俊辉

deng@tsinghua.edu.cn

# 桶排序：简单情况

❖ 给定[0, m)内的n个互异整数，如何高效地排序？ //必有n ≤ m

❖ 借助散列表E[0, m) //各元素仅需1个bit

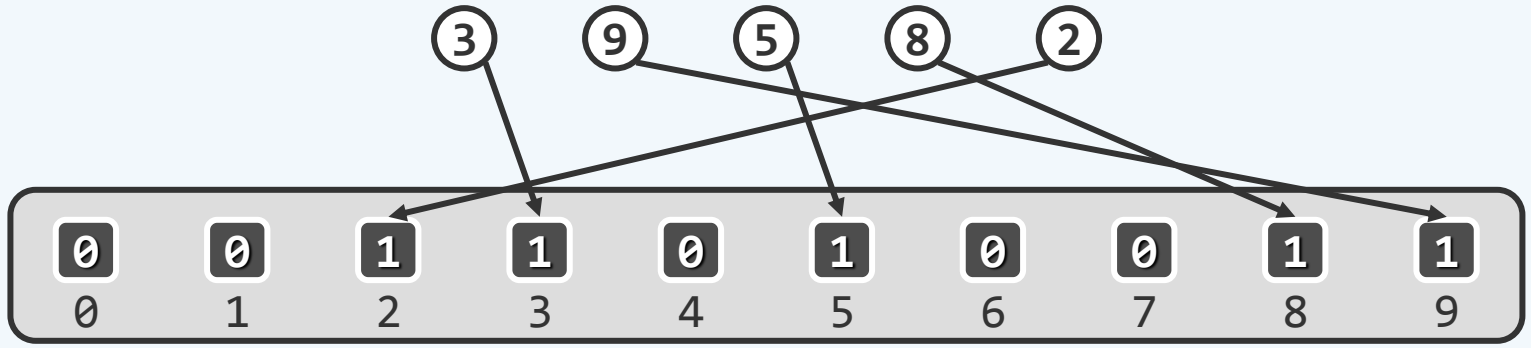
initialization　for i = 0 to m - 1, let E[i] = 0 //O(m)，可优化至O(1)

distribution　for each key in the input, let E[key] = 1 //O(n)

enumeration　for i = 0 to m - 1, output i if E[i] = 1 //O(m)

❖ 空间：O(m)

时间：O(n + m)



# 桶排序：一般情况

- 进一步地，若允许关键码重复 //此时未必n ≤ m，甚至可能m << n

  比如，清华大学2013级本科生按生日排序，则有n = 3300，m = 365

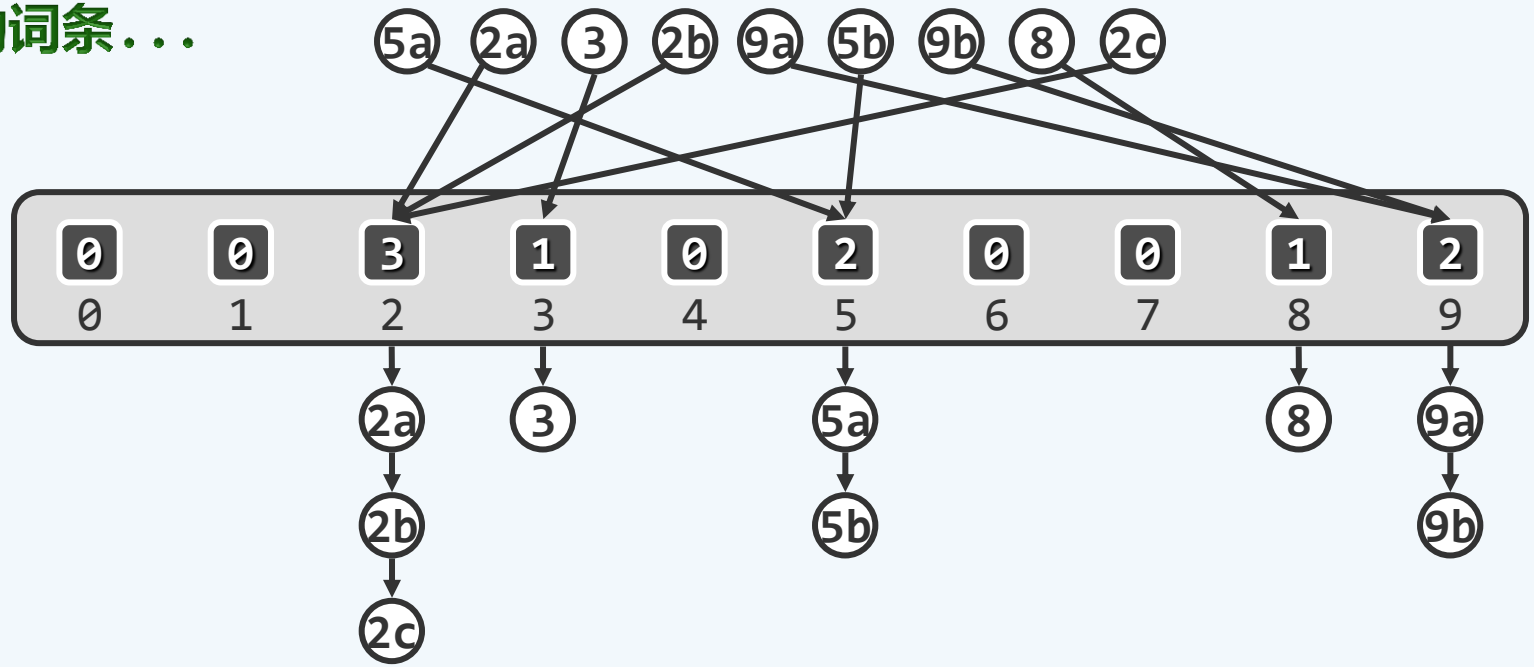
- 依然使用散列表，相互冲突的词条...

  组成独立链

- 空间复杂度

  = 散列表长 + 所有链表总长

  = O(m + n) //改用向量呢

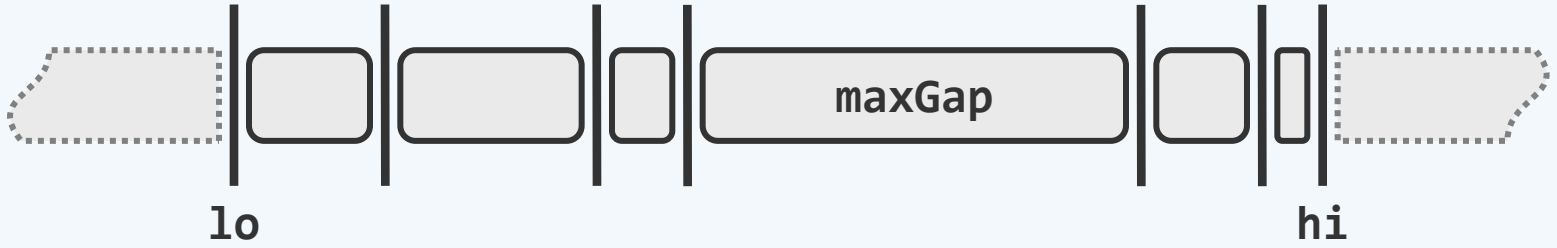


# 桶排序：一般情况

- initialization　初始化散列表（开辟空间、设置各桶的表头）　//如有必要，可以优化
  - distribution　扫描各词条，散列并插至对应桶的链表　//插入位置有讲究
  - collection　扫描各桶，串接所有非空链表　//串接次序和方向也有讲究
- 只要实现得当，必能保证稳定性，即雷同词条的次序与输入相同　//其重要性，远超直觉
- 时间复杂度 = $O(m) + O(n) + O(m) = O(n + m)$
- 大量词条重复时m << n，性能接近于线性
- 关键码均匀分布时，亦是如此

# MaxGap：平凡算法

❖ 任意n个互异点均将实轴分为n - 1段有界区间，其中哪一段最长？




❖ 平凡算法：对所有点排序 //最坏情况下Ω(nlogn)

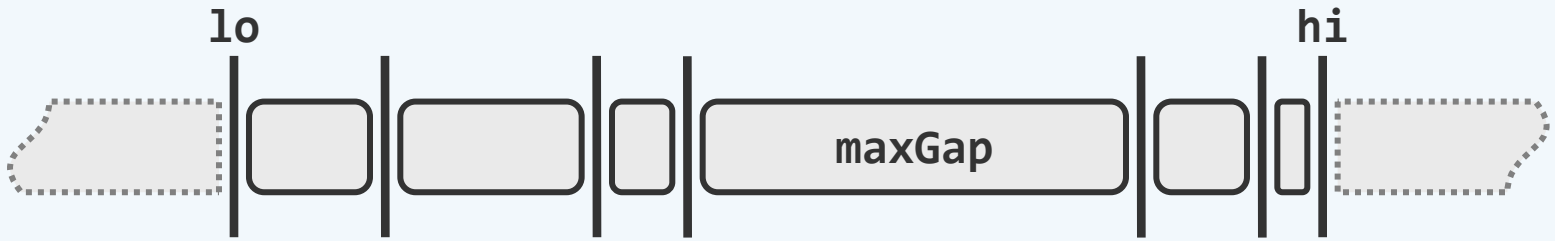
依次计算各相邻点对的间距，保留最大者 //Θ(n)

❖ 可否更快？

❖ 采用分桶策略，可改进至O(n)时间...

# MaxGap：线性算法

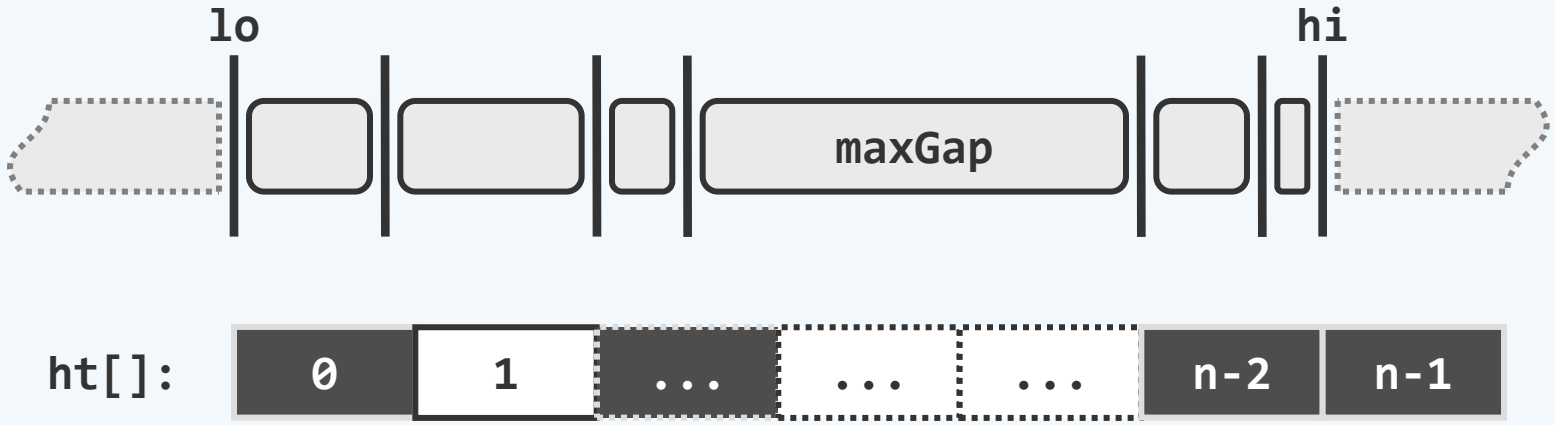
- 找到最左点、最右点 O(n) //一趟线性扫描
- 将有效范围均匀地划分为n-1段（桶） O(n) //相当于散列表
- 通过散列，将各点归入对应的桶 O(n) //模余法
- 在各桶中，动态记录最左点、最右点 O(n) //可能相同甚至没有
- 算出相邻（非空）桶之间的“距离” O(n) //一趟遍历足矣
- 最大的距离即MaxGap O(n) //画家算法




# MaxGap：正确性

❖ 正确性：MaxGap至少与相邻的两个桶相交

等价地，定义MaxGap的点不可能属于同一个桶




❖ 对称的MinGap问题：n – 1段有界区间中，何者最短？

可否沿用上法，以突破Ω(nlogn)下界？