

The background features several overlapping circles in various shades of blue, ranging from light cyan to a deep, vibrant blue. The circles are positioned on the left side of the frame, creating a modern, abstract design.

软件设计师

--数据结构基础

高级项目经理 任铄

QQ: 1530841586

第一章 数据结构基础

- 1.1 线性表
- 1.2 树和二叉树
- 1.3 图
- 1.4 排序
- 1.5 查找

一、查找

给定一个值 K ，在含有 n 个记录的文件中进行搜索，寻找一个关键字值等于 K 的记录，如找到则输出该记录，否则输出查找不成功的信息。

高级项目经理 任铄

QQ: 1530841586

向上人生路!

1、查找算法的优劣

用比较次数的平均值来评估算法的优劣，称为平均查找长度ASL。

$$ASL = \sum_{i=1}^n p_i c_i$$

高级项目经理 任铄

QQ: 1530841586

二、静态查找表

静态查找表的查找算法主要有：

1. 顺序查找（线性查找）
2. 折半查找（二分或对分查找）
3. 分块查找（索引顺序查找）

高级项目经理 任铄

QQ: 1530841586

1、顺序查找

用逐一比较的办法顺序查找关键字。

- 性能分析：

顺序查找平均查找长度为： $(n+1)/2$ ，时间效率为 $O(n)$

- 优点：算法简单、适应面广，对查找表的结构没有要求，无论记录是否按关键字有序排列均可使用。

- 缺点：在 n 值较大时，平均查找长度较大，查找效率较低。

高级项目经理 任铄

QQ: 1530841586

2、折半查找（二分查找）

先给数据排序，形成有序表，把待查数据值与查找范围的中间元素值进行比较，会有四种情况出现：

- 1) 待查找数值与中间元素值相等，返回中间元素值的索引。
- 2) 待查找数值比中间元素值小，则以整个查找范围的前半部分作为新的查找范围，执行 1)，直到找到相等的值。
- 3) 待查找数值比中间元素值大，则以整个查找范围的后半部分作为新的查找范围，执行 1)，直到找到相等的值
- 4) 如果最后找不到相等的值，则返回错误提示信息。

定义数组a[11]

找 key = 21

在有序的数列中查找

5 13 19 21 37 56 64 75 80 88 92

↑
Low

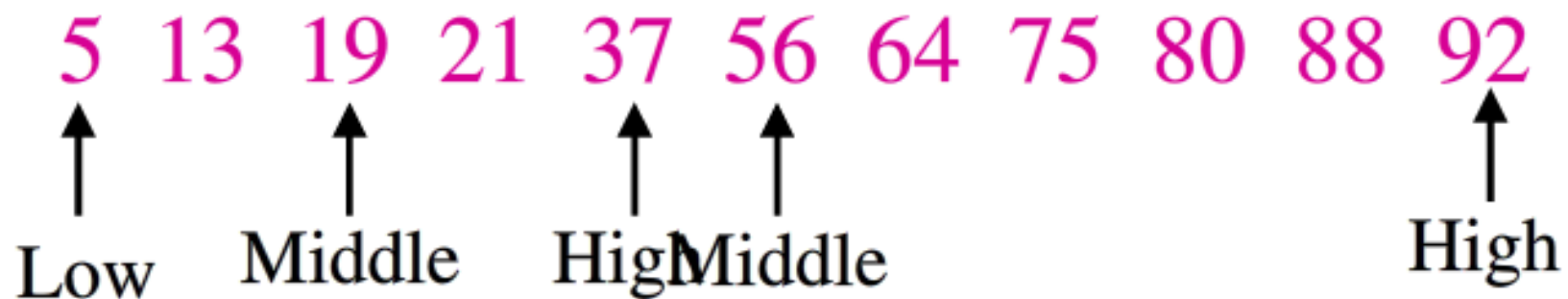
↑
Middle

↑
High

key < a[Middle]

定义数组a[11]

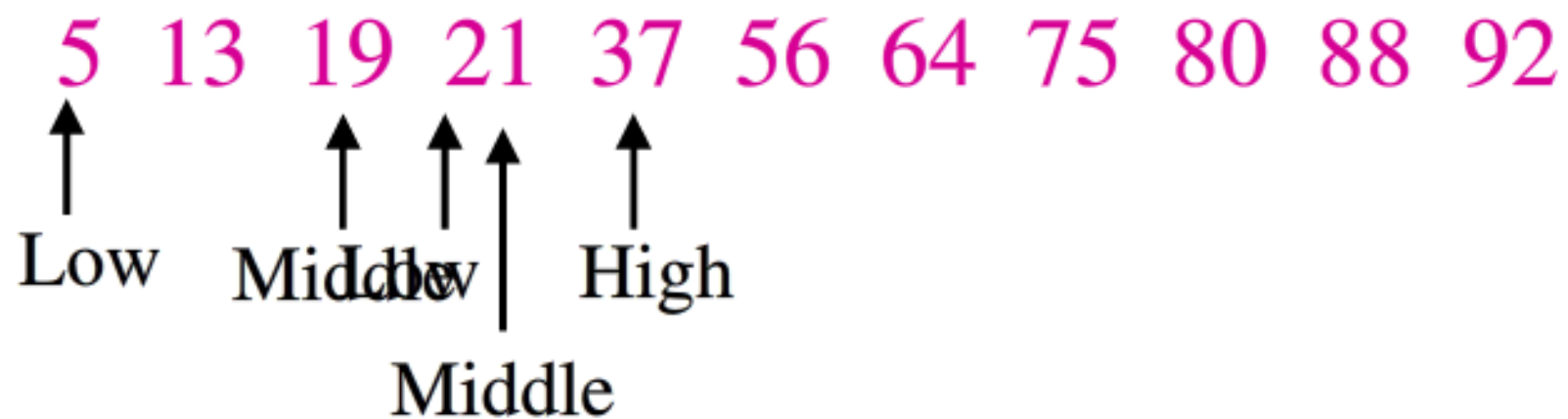
找 key = 21



key > a[Middle]

定义数组a[11]

找 key = 21



key = a[Middle] 找到

平均查找长度： $(n+1)/n \log_2(n+1)-1 \approx \log_2(n+1)-1$

折半查找比顺序查找的效率要高，但它要求查找表进行顺序存储并且按关键字有序排列，因此对表进行元素的插入和删除时，需要移动大量的元素，所以折半查找适用于表不易变动，且又经常进行查找的情况。

高级项目经理 任铄

QQ: 1530841586

向上人生路!

高级项目经理 任铄

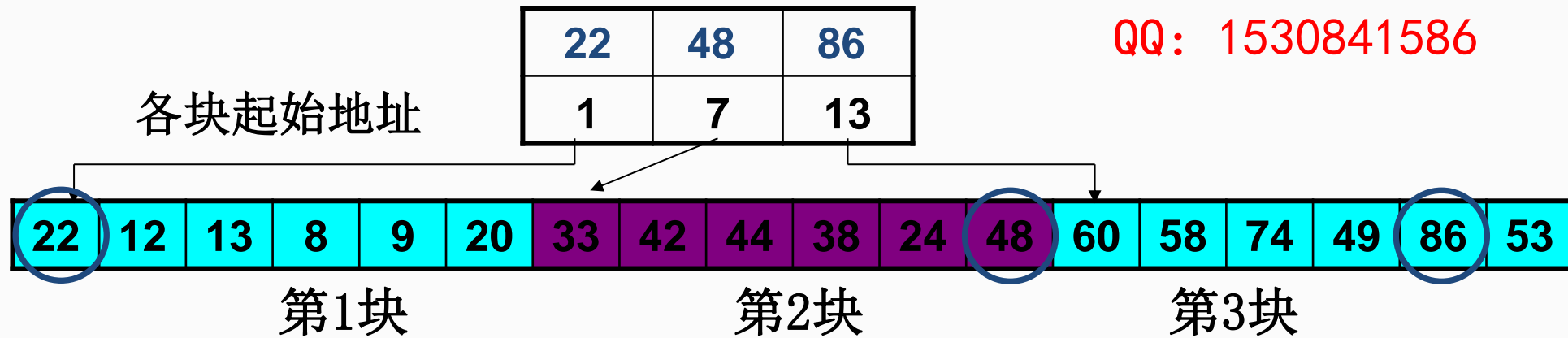
QQ: 1530841586

3. 分块查找

又称索引顺序查找，是对顺序查找方法的一种改进，其性能介于顺序查找和折半查找之间。

在分块查找过程中，首先把表分成若干块，每一块中的关键字不一定有序，但块之间是有序的，即后一块中所有记录的关键字均大于前一个块中最大的关键字，还建立了一个索引表，索引表按关键字有序。

高级项目经理 任铄
QQ: 1530841586



查找步骤分两步进行：

- ① 对索引表使用折半查找法（因为索引表是有序表）；
- ② 确定了待查关键字所在的子表后，在子表内采用顺序查找法（因为各子表内部是无序表）；

三、哈希表

是根据关键码值(Key value)而直接进行访问的数据结构。由关键码的值决定数据的存储地址，以加快查找的速度。这个映射函数叫做散列函数，存放记录的数组叫做散列表。

优点：查找速度极快（ $O(1)$ ），查找效率与元素个数 n 无关！

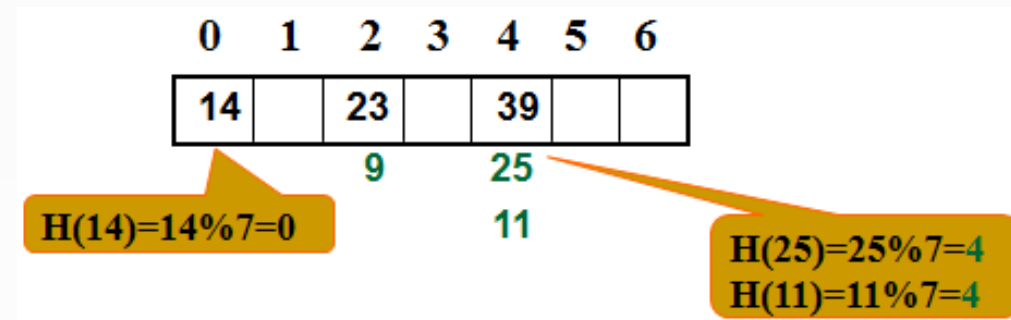
高级项目经理 任铄

QQ: 1530841586

通常关键码的集合比哈希地址集合大得多，因而经过哈希函数变换后，可能将不同的关键码映射到同一个哈希地址上，这种现象称为冲突。

有6个元素的关键码分别为：（14，23，39，9，25，11）。

选取关键码与元素位置间的函数为 $H(k)=k \bmod 7$



在哈希查找方法中，冲突是不可能避免的，只能尽可能减少。

1) 构造好的哈希函数

- 所选函数尽可能简单，以便提高转换速度；
- 所选函数对关键码计算出的地址，应在哈希地址内集中并大致均匀分布，以减少空间浪费。

2) 制定一个好的解决冲突的方案

查找时，如果从哈希函数计算出的地址中查不到关键码，则应当依据解决冲突的规则，有规律地查询其它相关单元。

可以通过下列渠道沟通联系：

- 1、QQ:1530841586
- 2、QQ群：164955673

向上人生路！