



王道计算机考研
www.cskaoyan.com

查找

www.cskaoyan.com 王道论坛

微信公众号：王道在线

王道论坛网址：www.cskaoyan.com



顺序查找

顺序查找又称线性查找，它对于顺序表和链表都是适用的。对于顺序表，可通过数组下标递归来顺序扫描每个元素；对于链表，则通过指针next来依次扫描每个元素



折半查找又称二分查找，它仅适用于**有序的顺序表**

。折半查找的基本思想：首先将给定值key与表中中间位置的元素比较，若相等，则查找成功，返回该元素的存储位置；若不等，则所需查找的元素只能在中间元素以外的前半部分或后半部分（例如，在查找表升序排列时，若给定值key大于中间元素，则所查找的元素只可能在后半部分）。然后在缩小的范围内继续进行同样的查找，如此重复，直到找到为止，或确定表中没有所需要查找的元素，则查找不成功，返回查找失败的信息

针对顺序表有序，我们使用qsort来排序，具体排序算法在第八章进行讲解，qsort的使用方法如下：

```
#include <stdlib.h> void qsort( void *buf,  
size_t num, size_t size, int (*compare)(const  
void *, const void *) );
```

buf:要排序数组的起始地址

num: 数组中元素的个数

size: 数组中每个元素所占用的空间大小

compare:比较规则，需要我们传递一个函数名



通过网址

<https://www.cs.usfca.edu/~galles/visualization/Search.html>来演示折半查找的动画效果



哈希查找 哈希即散列



散列函数：一个把查找表中的关键字映射成该关键字对应的地址的函数，记为 $\text{Hash}(\text{key}) = \text{Addr}$ （这里的地址可以是数组下标、索引或内存地址等）。

散列函数可能会把两个或两个以上的不同关键字映射到同一地址，称这种情况为**冲突**。

散列表：根据关键字而直接进行访问的数据结构。也就是说，散列表建立了关键字和存储地址之间的一种直接映射关系。

理想情况下，对散列表进行查找的时间复杂度为 $O(1)$ ，即与表中元素的个数无关



```
#define MaxKey 1000
```

```
int hash(char* key)
{
    int h=0,g;
    while(*key)
    {
        h=(h<<4)+*key++;
        g=h&0xf0000000;
        if(g)
        {
            h^=g>>24;
        }
        h&=~g;
    }
    return h%MaxKey;
}
```




```
void use_hash()
{
    char *pStr[5]={"xiongda","lele","hanmeimei","wangdao","fenghua"};
    int i;
    char *pHash_table[MaxKey]={NULL};
    for(i=0;i<5;i++)
    {
        printf("%s is key=%d\n",pStr[i],hash(pStr[i]));
        pHash_table[hash(pStr[i])]=pStr[i];
    }
}
```

KMP—理解原理即可， 不要求手撸代码，不会 考大题，只会考选择题

是王道数据结构的第四章，内容不多，主要是KMP，因此放到这里讲解

比对某个小字符串是不是
在一个很长的字符串中出
现过，首先咱们来看暴力
比对的代码



KMP原理

(如图2所示) 当c与b不匹配时, 已匹配 'abca' 的前缀a和后缀a为最长公共元素。已知前缀a与b、c均不同, 与后缀a相同, 故无须比较, 直接将子串移动“已匹配的字符数 - 对应的部分匹配值”, 用子串前缀后面的元素与主串匹配失败的元素开始比较即可

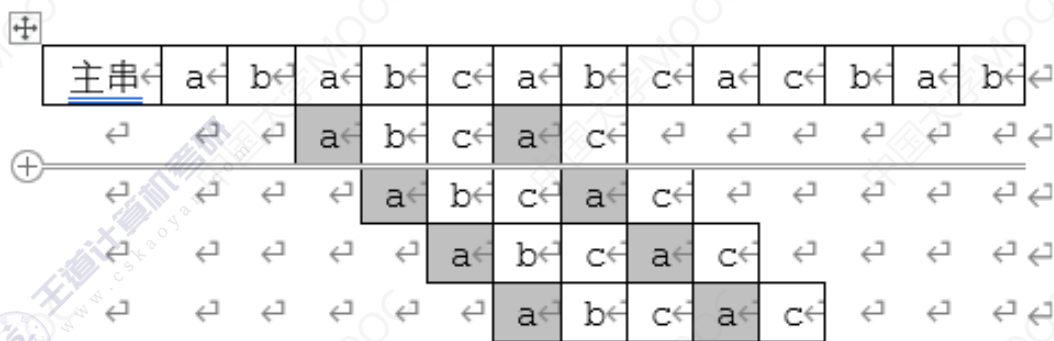


图1 失配后移动情况 (暴力)

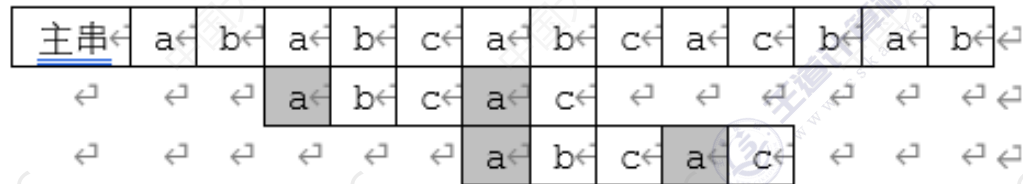


图2 直接移动到合适位置



next数组

编号	1	2	3	4	5
s	a	b	c	a	c
next	0	1	1	1	2

KMP理解原理即可，只会出选择题