

10.1.1 算法概述

10.1.2 选主元

1. 随机取，rand()函数耗时太长了
2. 取头中尾的中位数
 1. 例如：8，12，3的中位数就是8
 2. 测试一下pivot不同的取法对运行速度有多大影响

```
ElementType Median3(ElementType A[],int Left,int Right)
{
    int Center=(Left+Right)/2;
    if(A[Left]>A[Center])
        Swap(&A[Left],&A[Center]);
    if(A[Left]>A[Right])
        Swap(&A[Left],&A[Right]);
    if(A[Center]>A[Right])
        Swap(&A[Center],&A[Right]);
    /* A[Left]<=A[Center]<=A[Right] */
    Swap(&A[Center],&A[Right-1]); /* 将pivot藏到右边 */
    /* 只需要考虑A[Left+1]...A[Right-2] */
    return A[Right-1];
}
```

10.1.3 子集划分

如果有元素恰好等于pivot

停下来交换！

快速排序的问题

1. 用递归
2. 对小规模的数据，可能还不如插入排序快 解决方案
3. 当递归的数据规模充分小，则停止递归，直接调用简单排序
4. 在程序中定义一个Cutoff的阈值

10.1.4 算法实现
