

简单算法

二分查找

顺序查找比较简单，时间复杂度是 $O(n)$

任务：给定一个 n 个元素有序（升序）整型数组`nums`和一个目标值`target`，写一个函数搜索`nums`中的`target`，如果目标值存在返回下标，否则返回`-1`

动画示意：[快挂](#)

```
int *binsearch(int x,int *v,int n)
{
    int *low = &v[0];
    int *high = &v[n];
    int *mid = NULL;
    while(low<high){    //如果数组中不存在目标值的话，重复查找会使low>high
        mid = low + (high-low)/2    //注意！！指针不能相加
        if(x < *mid)
            high = mid;
        if(x > *mid)
            low = mid + 1;
        else
            return mid;
    }
    return NULL;
}
```

冒泡排序

```
def bubble_sort(a[n]):

    boolean flag = false;    //定义一个标志位，判定元素之间是否进行了交换
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<i-1;j++)
        {
            if(a[j]>a[j+1])    //进入这个if分支里，说明有元素进行了交
换
            {
                flag=true;    //将flag=true
                int temp = a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
```

```
        // 在进行完一轮的排序后，判断本轮是否发生了元素之间的交换
        // 如果没有，说明数组有序，直接结束
        if(!flag){
            break;
        } else{
            // 如果发生了交换，那么下一轮排序前将flag再次置换成true
            // 以便记录下一轮排序时是否会发生交换
            flag=false;
        }
    }
}
```

时间复杂度是 $O(n^2)$