习题1

* 1. 选择排序

1.1.1.算法描述

选择排序（Selection sort）是一种简单直观的[排序算法](https://baike.baidu.com/item/%E6%8E%92%E5%BA%8F%E7%AE%97%E6%B3%95/5399605" \t "https://baike.baidu.com/item/%E9%80%89%E6%8B%A9%E6%8E%92%E5%BA%8F/_blank)。它的工作原理是：第一次从待排序的[数据元素](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0/715313" \t "https://baike.baidu.com/item/%E9%80%89%E6%8B%A9%E6%8E%92%E5%BA%8F/_blank)中选出最小（或最大）的一个元素，存放在序列的起始位置，然后再从剩余的未排序元素中寻找到最小（大）元素，然后放到已排序的序列的末尾。以此类推，直到全部待排序的数据元素的个数为零。选择排序是不稳定的排序方法。

1.1.2.伪代码

chooseSort(int\* a,int len)

//使用chooseSort排序

//输入：数组a

//输出：数组a排序后的数字

for i 1 to len-1

max=i

for j i to len-1

if max < a[j]

max=a[j]

交换max i

1.1.3.代码实现（C语言）

chooseSort(int\* a,int len){

int max,tmp;

for(int i=0;i<len-1;i++){

max=i;

for(int j=i;j<len;j++)

if(a[max]<a[j]) max=j;

tmp=a[max];

a[max]=a[i];

a[i]=tmp;

}

}

* 1. **冒泡排序**

1.2.1.算法描述

冒泡排序，从下往上进行大数起泡，起n次即可达到排序效果。

1.2.2.伪代码

bubbleSort(int\* a,int len)

//使用bubbleSort排序

//输入：数组a

//输出：数组a排序后的数字

for i 1 to len

for j len to i

if(a[j]>a[j-1])

交换a[j]和a[j-1]

1.2.3.算法实现（C语言）

bubbleSort(int\* a,int len){

int tmp;

for(int i=1;i<len;i++){

for(int j=len-1;j>i;j--){

if(a[j]>a[j-1]){

tmp=a[j];

a[j]=a[j-1];

a[j-1]=tmp;

}

}

}

}

* 1. **试验小结**

关于冒泡排序和选择排序，时间复杂度都是O(n^2)，空间复杂度都是O(1)，所以，我们进行比较这两种排序方法是否优越，完全看稳定性。关于选择排序，是不稳定的，而冒泡排序是稳定的，所以，我们尽可能不要使用chooseSort。但是选择排序也有优势，比如，在一个序列中，少量的数字趋于无序，那么，也可以使用选择排序。