

计算机等级考试二级辅导c 常用排序算法 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/133/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E7_c97_133699.htm //选择排序

```
法SelectionSort(int arr[],int n) template void SelectionSort(T
arr[],int n) { int smallIndex. //表中最小元素的下标 int pass,j //用
来扫描子表的下标 T temp. //用来交换表元素的临时变量 //pass
的范围是0~n-2 for (pass=0;pass<n-1;pass++) { //从下标pass开始扫描子表
smallIndex=pass. //j遍历整个子表arr[pass+1]到arr[n-1] for(j=pass
+1;j<n;j++) //如果找到更小的元素,则将该位置赋值给smallIndex if(arr[j]
<arr[smallIndex]) smallIndex=j. //如果smallIndex和pass不在相同的位置 //则将子
表中的最小项与arr[pass]交换 if(smallIndex!=pass) {
temp=arr[pass]. arr[pass]=arr[smallIndex]. arr[smallIndex]=temp. }
}}
```

/*

***** 双端选择排序算法:是上面选择排序算法的变种,可以定位每个子表中最小和最大元素 并把它们分别放在子表的开头和结尾.

```
*****//双端选择排序算法函数deSelSort()的实现 template
void deSelSort(T arr[],int n) { int smallIndex,largeIndex. //表中最
小及最大元素的下标 int leftPass=0,rightPass=n-1,i,j.//用来从表
左边及右边扫描子表的下标 T temp. //用于交换元素的临时变
量 while (leftPass<rightPass) { //从左边及右边开始扫描子表
smallIndex=leftPass. largeIndex=rightPass. //j和i遍历整个子
表arr[leftPass]~arr[rightPass] for (i=leftPass+1;i<rightPass;i++) //如果找到更小的
```

元素,则将该位置赋值给smallIndex if (arr[smallIndex]=i. //如果smallIndex和leftPass不在相同的位置 //则将子表中的最小项与arr[leftPass]交换 if (smallIndex!=leftPass) { temp=arr[leftPass]. arr[leftPass]=arr[smallIndex]. arr[smallIndex]=temp. } for (j=rightPass-1.j>leftPass.j--) if(arr[j]>arr[largeIndex]) largeIndex=j.

100Test 下载频道开通 , 各类考试题目直接下载。详细请访问 www.100test.com