

用Java实现几种常见的排序算法 PDF转换可能丢失图片或格式
， 建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022__E7_94_A8Java_E5_AE_9E_c97_138400.htm 用Java语言实现的各种排序，包括插入排序、冒泡排序、选择排序、Shell排序、快速排序、归并排序、堆排序、SortUtil等。

插入排序：

```
package org.rut.util.algorithm.support; import org.rut.util.algorithm.SortUtil; /** * @author treeroot * @since 2006-2-2 * @version 1.0 */ public class InsertSort implements SortUtil.Sort{ /* (non-Javadoc) * @see org.rut.util.algorithm.SortUtil.Sort#sort(int[]) */ public void sort(int[] data) { int temp; for(int i=1; i for(int j=i; (j>0)amp.(data[j] SortUtil.swap(data,j,j-1). } } } }
```

冒泡排序：

```
package org.rut.util.algorithm.support; import org.rut.util.algorithm.SortUtil; /** * @author treeroot * @since 2006-2-2 * @version 1.0 */ public class BubbleSort implements SortUtil.Sort{ /* (non-Javadoc) * @see org.rut.util.algorithm.SortUtil.Sort#sort(int[]) */ public void sort(int[] data) { int temp; for(int i=0; i for(int j=data.length-1; j>i; j--){ if(data[j] SortUtil.swap(data,j,j-1). } } } }
```

选择排序：

```
package org.rut.util.algorithm.support; import org.rut.util.algorithm.SortUtil; /** * @author treeroot * @since 2006-2-2 * @version 1.0 */ public class SelectionSort implements SortUtil.Sort { /* * (non-Javadoc) * * @see org.rut.util.algorithm.SortUtil.Sort#sort(int[]) */ public void sort(int[] data) { int temp; for (int i = 0; i int lowIndex = i. for (int j = data.length - 1; j >i; j--) { if (data[j] lowIndex = j. } } SortUtil.swap(data,i,lowIndex). } } }
```

Shell排序：

```
package
```

```

org.rut.util.algorithm.support. import org.rut.util.algorithm.SortUtil.
/** * @author treeroot * @since 2006-2-2 * @version 1.0 */ public
class ShellSort implements SortUtil.Sort{ /* (non-Javadoc) * @see
org.rut.util.algorithm.SortUtil.Sort#sort(int[]) */ public void
sort(int[] data) { for(int i=data.length/2;i>2;i/=2){ for(int j=0.j
insertSort(data,j,i). } } insertSort(data,0,1). } /** * @param data *
@param j * @param i */ private void insertSort(int[] data, int start,
int inc) { int temp. for(int i=start inc.i for(int
j=i.(j>=inc)amp.(data[j] SortUtil.swap(data,j,j-inc). } } } } 快速排
序 : package org.rut.util.algorithm.support. import
org.rut.util.algorithm.SortUtil. /** * @author treeroot * @since
2006-2-2 * @version 1.0 */ public class QuickSort implements
SortUtil.Sort{ /* (non-Javadoc) * @see
org.rut.util.algorithm.SortUtil.Sort#sort(int[]) */ public void
sort(int[] data) { quickSort(data,0,data.length-1). } private void
quickSort(int[] data,int i,int j){ int pivotIndex=(i j)/2. //swap
SortUtil.swap(data,pivotIndex,j). int k=partition(data,i-1,j,data[j]).
SortUtil.swap(data,k,j). if((k-i)>1) quickSort(data,i,k-1). if((j-k)>1)
quickSort(data,k 1,j). } /** * @param data * @param i * @param j *
@return */ private int partition(int[] data, int l, int r,int pivot) { do{
while(data[ l] while((r!=0)amp.data[--r]>pivot).
SortUtil.swap(data,l,r). } while(l SortUtil.swap(data,l,r). return l. } }
改进后的快速排序 : package org.rut.util.algorithm.support.
import org.rut.util.algorithm.SortUtil. /** * @author treeroot *
@since 2006-2-2 * @version 1.0 */ public class ImprovedQuickSort
implements SortUtil.Sort { private static int

```

```

MAX_STACK_SIZE=4096. private static int THRESHOLD=10. /*
(non-Javadoc) * @see org.rut.util.algorithm.SortUtil.Sort#sort(int[])
*/ public void sort(int[] data) { int[] stack=new
int[MAX_STACK_SIZE]. int top=-1. int pivot. int pivotIndex,l,r.
stack[ top]=0. stack[ top]=data.length-1. while(top>0){ int
j=stack[top--]. int i=stack[top--]. pivotIndex=(i+j)/2.
pivot=data[pivotIndex]. SortUtil.swap(data,pivotIndex,j).
//partition l=i-1. r=j. do{ while(data[ l]
while((r!=0)&(data[--r]>pivot)). SortUtil.swap(data,l,r). }
while(l SortUtil.swap(data,l,r). SortUtil.swap(data,l,j).
if((l-i)>THRESHOLD){ stack[ top]=i. stack[ top]=l-1. }
if((j-l)>THRESHOLD){ stack[ top]=l+1. stack[ top]=j. } } //new
InsertSort().sort(data). insertSort(data). } /** * @param data */
private void insertSort(int[] data) { int temp. for(int i=1;i for(int
j=i.(j>0)&(data[j] SortUtil.swap(data,j,j-1). } } } } 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

```