Java 算法

作者：xixiaoming

原文链接：https://ld246.com/article/1541468238223

来源网站：链滴

排序算法

排序算法分类

我们这里写的都只是内部排序，即只使用内存的排序算法

1. 插入排序

1.1 直接插入排序

1.2 希尔排序

2. 选择排序

2.1 简单选择排序

2.2 堆排序（所需辅助空间最小）

3. 交换排序

3.1 冒泡排序

3.2 快速排序（平均速度最快）

4. 归并排序（所需辅助空间最多）

5. 基数排序

排序算法选择

1. 当数据规模较小---->直接插入排序或冒泡排序

2. 如果全部为正整数---->桶排序

3. 我们说的快排最好是指大量随机数据下，快排的平均时间复杂度最好

4. 稳定的排序方法指的是，对于两个关键字相等的记录，它们在序列中的相对位置，在排序之前和经排序之后，没有改变。其中冒泡排序和归并排序是稳定的，希尔排序和堆排序是不稳定的。

Java算法例子

斐波那契数列

```
/**
 * 有一对兔子，从出生后第3个月起每个月都生一对兔子，
 * 小兔子长到第三个月后每个月又生一对兔子，
 * 假如兔子都不死，问每个月的兔子总数为多少?
 * @param month 第几个月
 * @return 当月兔子总数
 */
public Integer rabbit(int month){
    // 数字规律1,1,2,3,5,8,13,21....后面一个永远是前面两个之和，也就是斐波那契数列
    if(month==1||month==2){
        return 1;
    }else{
        return rabbit(month-1)+rabbit(month-2);
    }
```

```
}
```

水仙花数

```
/**
 * 打印出所有的"水仙花数"，所谓"水仙花数"是指一个三位数，其各位数字立方和等于该数本身。例
:
   * 153是一个"水仙花数"，因为153=1的三次方＋5的三次方＋3的三次方
 */
public void daffodilNumber(){
    for(int i=100;i<1000;i++){
        int i1 = i/100;
        int i2 = (i-i1*100)/10;
        int i3 = i-i1*100-i2*10;
        if(i==i1*i1*i1+i2*i2*i2+i3*i3*i3){
            System.out.print(i+",");
        }
    }
}
```

分解质因数

```
/**
 * 将一个正整数分解质因数
 * 例如：输入90,打印出90=2*3*3*5
 * @param num 要分解的正整数
 * @return 分解结果
 */
public String resolveNumber(int num){
    String result = num + "=";
    int i = 2;
       while (i<num) {
          if (num % i == 0) {
           result+=i + "*";
              num = num / i;
          } else {
              i++;
          }
       }
       return result+=i;
}
```

最大公约数和最小公倍数

```
/**
 * 输入两个正整数，求其最大公约数和最小公倍数
 * @param a,b 输入的两个正整数
 */
public void comonDivisor (int a,int b){
```

```
    for(int i=a>b?b:a;i>=1;i--){
        if(a%i==0 && b%i==0){
            System.out.println("最大公约数"+i);
            System.out.println("最小公倍数"+a*b/i);
            return;
        }
    }
}
```

完数

```
/**
 * 一个数如果恰好等于它的因子之和，这个数就称为"完数"
 * 例如6=1＋2＋3.编程 找出10000以内的所有完数
 */
public void wanShu (){
    for(int index=1;index<=10000;index++){
        int sum = 1,i=2,temp = index;
        while(i<=temp){
            if(temp%i==0){
                sum+=i;
                temp=temp/i;
            }else{
                i++;
            }
        }
        if(sum==index){
            System.out.print(index+",");
        }
    }

}
```

计算年度第几天

```
/**
 * 输入某年某月某日，判断这一天是这一年的第几天
 */
public int day(int year,int month,int day){
    int date = 0;
       int arr[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
       if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0)) {//闰年
           arr[1] = 29;
       }
       for (int i = 0; i < month - 1; i++) {
           date += arr[i];
       }
       date += day;
       return date;
}
```

序列求和

```
/**
 * 有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13...
 * 求出这个数列的前num项之和
 */
public Double add(int num){
    Double i1=1d;
    Double i2=2d;
    Double result = 0d;
    for(int i=0;i<num;i++){
        result+=(i2/i1);
        i2 = i1+i2;
        i1 = i2-i1;
    }
    return result;
}
```

阶乘

```
/**
 * 求n!，也就是1*2*3*4乘到n过
 */
public int multiplicate(int n){
    if(n==1){
        return 1;
    }else{
        return n*multiplicate(n-1);
    }
}
```

素数

```
/**
 * 求n之内的素数
 */
public void primeNumber(int n){
    int j = 0 ;
    boolean isprime = true;
    for(int i=2;i<=n;i++){
        j = (int) (Math.sqrt(i));
        for(int m=2;m<j;m++){
            if(i%m==0){
                isprime = false;
                break;
            }
        }
        if(isprime){
            System.out.print(i+",");
        }
        isprime = true;
```

```
    }
}
```

杨辉三角形

```
/**
 * 打印出杨辉三角形
 */
public void yanghui(int line) {
    int[] a = new int[line];
    for (int i = 0; i < line; i++) {
        a[i] = 1;
    }
    if (line == 1) {
        System.out.println(1);
    } else if (line == 2) {
        System.out.println(1);
        System.out.println(1 + "\t" + 1);
    } else {
        System.out.println(1);
        System.out.println(1 + "\t" + 1);
        for (int i = 1; i < line - 1; i++) {
            for (int j = i; j >= 1; j--) {
                a[j] = a[j] + a[j - 1];
            }
            for (int k = 0; k < i + 2; k++) {
                System.out.print(a[k] + "\t");
            }
            System.out.println();
        }
    }
}
```

围圈报数

```
/**
* 有n个人围成一圈，顺序排号。
 * 从第一个人开始报数（从1到3报数），凡报到3的人退出圈子，
 * 问最后留下的是原来第几号的那位
 */
public void quit(int n) {
    int[] a = new int[n];
      int i = 0;
      int t = 0;
      while (n > 1) {
          if (a[i] == 0) {
              t++;
              if (t == 3) {
                  t = 0;
                  a[i] = 1;
                  n--;
```

```
            }
        }
        i++;
        if(i == a.length){
            i = 0;
        }
    }
    for(int j=0;j<a.length;j++){
        if(a[j]!=1){
            System.out.println(j+1);
        }
    }
}
```

一个偶数总能表示为两个素数之和

```
public void even(int num) {
    int j = 0, num2 = 0, flag = 0, tag = 0, temp = 0;
    for (int i = 3; i <= num/2; i++) {
        j = (int) Math.sqrt(i);
        for (int k = 2; k <= j; k++) {
            if (i % k == 0) {
                flag = 1;
            }
        }
        if (flag == 0) {
            num2 = num - i;
            temp = (int) Math.sqrt(num2);
            for (int k = 2; k <= temp; k++) {
                if (num2 % k == 0) {
                    tag = 1;
                }
            }
            if (tag == 0 && num2 >= 3) {
                System.out.println(num + "=" + i + "+" + num2);
            }
            tag = 0;
        }
        flag = 0;
    }
}
```

选择题

1. 下列数据结构具有记忆功能的是

A．队列

B．循环队列

C．栈

D．顺序表

栈先进后出。最先进去的数肯定是最后出来的 所以说有记忆功能。队列是先进先去，如果进行出队操

，最先的元素就出队没有了，没有记忆功能。

2. 在选项中，只要指出表中任何一个结点的位置，就可以从它出发依次访问到表中其他所有结点。

A．线性单链表

B．双向链表

C．线性链表

D．循环链表

简答题

heap （堆）和stack（栈）有什么区别

栈是一种线形集合，其添加和删除元素的操作应在同一段完成，按照后进先出的方式进行处理；

堆是栈的一个组成元素。

JAVA中引用及原始类型变量都放在栈中，new出来的对象放在堆中

栈和队列的共同特点是什么

只允许在端点处插入和删除元素

栈通常采用的两种存储结构是什么

线性存储、链表存储

用链表表示线性表的优点是什么

便于插入和删除操作

在单链表中，增加头结点的目的是

方便运算的实现

循环链表的主要优点是什么

从表中任一结点出发都能访问到整个链表

树是结点的集合，它的根结点数目是多少

有且只有 1

在深度为 5 的满二叉树中，叶子结点的个数为

2的4次方=16

设一棵二叉树中有 3 个叶子结点，有8 个度为 1 的结点，则

二叉树中总的结点数为多少

2*3+8-1=13

算法是指什么，他的四个基本特征，可以用哪几种控制结构成

算法是解题方案的准确而完整的描述，他有四个特征：可行性、确定性、有穷性和拥有足够的情报，般都可以用顺序、选择、循环等控制结构组合而成

算法的时间、空间复杂度是指

算法执行过程中所需要的基本运算次数、存储空间

算法分析的目的是

分析算法的效率以求改进

数据的存储结构是指什么

数据的逻辑结构在计算机中的表示

数据的逻辑结构是指

反映数据元素之间逻辑关系的数据结构

设一棵完全二叉树共有 699 个结点，则在该二叉树中的叶子点数为

n/2向上取整=350

根据数据结构中各数据元素之间前后件关系的复杂程度，一将数据结构分为

线性结构和非线性结构

串的长度是

串中所含字符的个数

设有两个串 p 和q，求q 在p 中首次出现位置的运算称做

模式匹配