

解析法与问题解决

用 Python 编程通过解析式求解数学函数

基本思想:



在解析算法的程序实现过程中，首先要确保**数学表达式**的正确性，然后在程序中正确描述该数学表达式。

任务：判断一个数是否是**3**和**7**的公倍数

(1) 抽象与建模

◆找到核心要素

◆得出计算模型

假设要判断的数为**x**
判断条件:**3**和**7**的公倍数

3和**7**的公倍数
 $x\%3==0$ and $x\%7==0$

x是公倍数 或
x不是公倍数

(2) 设计算法

◆输入数据

◆处理数据

◆输出数据

输入**X**

$x\%3==0$ and $x\%7==0$

输出**x**是公倍数或**x**不是公倍数

(3) 编写程序

```
x=int(input())
if x%3==0 and x%7==0:
    print(x,"是公倍数")
else:
    print(x, "不是公倍数")
```

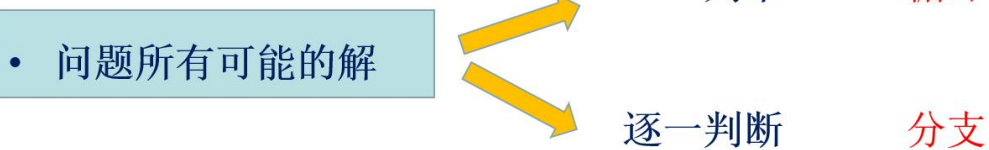
21

21 是公倍数

枚举法与问题解决

枚举算法思想是把所有可能解一一列举，然后判断每一个列举出的可能解是否为正确解。

基本思想：



在枚举算法中，**逐一列举**出每一个可能解，判断其是否为正确解的过程采用**循环**结构来实现；而在利用问题提供的约束条件**判断正确解**的过程中，则需要用到**分支**结构。

在设计枚举算法时，不能遗漏任何一个正确解，又要尽可能地缩小列举范围，以提高算法的工作效率。

用 Python 编程通过枚举法求解数学函数

File Edit Format Run Options Window Help

```
x=1
y=1
z=1
for x in range(1, 11):
    for y in range(1, 9):
        for z in range(1, 7):
            if 4*x+5*y+6*z==50:
                print(x, y, z)
```

枚举法求解 3 元 1 次方程

