

埃及分数（贪心算法简单示例）

1. 问题

埃及分数是指分子是1的分数，也叫单位分数。古代埃及人在进行分数运算时，只使用分子是1的分数。因此这种分数也叫做埃及分数，或者叫单分子分数。

给定一个分数，如7/8，我们可以把它表示为1/2 + 1/3 + 1/24，埃及分数问题即把一个真分数表示为最少的埃及分数之和的形式。

2. 贪心算法

贪心算法（又称贪婪算法）是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的是在某种意义上的局部最优解

由于贪心算法的每次都是贪婪选择的特性，我们可以用7/8来举例，小一点的埃及分数是怎么算出来的，你完全可以举例，前提是埃及分数必须要 1/x 的形式。

- 比7/8小一点的埃及分数 是多少，应该是 1/2 （ 4/8 ），去除1/2后剩3/8。
- 比3/8小一点的埃及分数是 1/3 （ 3/9 ），去除之后剩 1/ 24 ，得到最终答案。

那么该如何用代码实现呢？

3. 如何实现

下面最核心的问题应该是：

如何找到真分数包含的最大埃及分数？

设真分数 a/b，b 除以 a得整数部分为c，余数部分d。
那么按照数学上的运算，可以得到如下等式：

b = a * c + d [1]

凭空冒出这么个东西还真不习惯。
拿7/8举例子。a = 7，b = 8
那么 按照[1]式 8 = 7 * 1 + 1

我们对[1]式进行运算，两边同除a，得到[2]式

b / a = (a * c + d) / a [2]

对[2]式右边进行化简得到[3]

b/a = c + d/a [3]

由前提条件：d是a得余数，那么a肯定要比d大。（小学数学问题，不啰嗦了）
那么可以得到下面的[4]

b/a = c + d/a < c + 1 [4]

我们对两边取倒数，即可得到如下的[5] (注意取到数的符号变化)

a/b > 1 / (c + 1)[5]

已经可以看出 比 a / b 小的埃及分数了，且 1 / (c + 1) 一定是 a / b所包含的最大埃及分数。

已经可以证明1/(c +1) 是 a/b的埃及分数了，但为什么说1/(c + 1)是a/b 所包含的最大埃及分数？

其实可以由[4] 得出 a/b = 1 / (c + d/a) ，而d/a 一定小于1，我们要找最大的埃及分数，就要取一个最小的整数分母，而最小（最接近）的整数分母就是 c + 1了，所以1/(c + 1) 一定是其包含的最大的埃及分数了

这个思想有点逆乎常人，是一种类似反推的想法，但是知道能算出来就OK了。

下面我们设 e = c + 1(注意，1/e是最大埃及分数)
按照上面我们所说，一个真分数减去它的最大埃及分数。
运算出来就是：

a / b - 1 / e ** [6]

我们把它通分。

$$\frac{a}{b} - \frac{1}{e} = \frac{a * e}{b * e} - \frac{1 * b}{e * b} = \frac{a * e - b}{b * e}$$

通分后的结果

我们就可以知道 一个真分数减去一个最大埃及分数之后
原来的a 变成了，a * e - b，原来的b变成了 b * e

那么下面来看看核心代码

```
1 // 如果如果分子大于1就可以拆分
2 do {
3     e= b/a + 1;
4     System.out.println("1/"+e);
5
6     a = a * e- b;
7     b = b * e;
8
9     int maxDiv = maxConDiv(a,b);
10    if(maxDiv > 1){
11        a /= maxDiv;
12        b /= maxDiv;
13    }
14 }while (a > 1);
```

最外层的do-while循环好理解，如果a（分子）不是1，那么就不是埃及分数，就要一直拆分。
拆分的过程我刚才也已经说过了，分为下面3步。

- 找到当前真分数的最大埃及分数**
刚才我们说过，最大的埃及分数是 1 / (c + 1) 又因为e = c + 1，所以 1/e = 1 / (c + 1)，而我们目的是求当前真分数的最大埃及分数，所以也就是求 1/e，进而推要求e，e = b/a + 1，有同学可能会问，d哪里去了，根据[4]式，b/a = e - 1 + d/a ==> e = b/a + d/a -1,完全不一样啊？
别忘了我们是在写程序，b/a只会求得正数部分c，而不会求得余数部分d。

按照java得原则，
[1]式 就是 b = a * c
那自然就有 b / a = e - 1
即 e = b / a + 1

那要按照某些不损失精度的语言来说，可就要按照上面走啦。

- 减去最大埃及分数后通分**
上面的通分图片中我们已经看到了， a / b 减去最大埃及分数后通分的样子，所以直接写代码：

```
1 a = a * e- b;
2 b = b * e;
```

- 约分**
通完分之后又要约分，这是什么操作？
因为我们不约分就又整出更大的分数来了，这样运算更麻烦，可能还求不出来

经测试，不约分会产生死循环，根本求不出目标答案
约分是怎么个操作，首先找到最大公约数r，然后a/r，b/r即可完成约分，那么最大公约数必然想到用辗转相除法来求，最后面将补充辗转相除的算法。

```

1 import java.io.BufferedReader;
2 import java.io.InputStreamReader;
3
4 //埃及分数
5 public class EgyptFraction {
6     public static void main(String[] args) throws Exception{
7         BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
8         System.out.println("输入分子");
9         int a = Integer.parseInt(bufferedReader.readLine());
10        System.out.println("输入分母");
11        int b = Integer.parseInt(bufferedReader.readLine()); //输入ab
12
13        int e;
14
15        //要求是如果分子大于1就可以拆分
16        do {
17            e = b/a + 1;
18            System.out.println("1/"+e);
19
20            a = a * e - b;
21            b = b * e;
22
23            int maxDiv = maxComDiv(a,b);
24            if(maxDiv > 1){
25                a /= maxDiv;
26                b /= maxDiv;
27            }
28            while (a > 1);
29            System.out.println("1/"+b);
30        }
31
32        private static int maxComDiv(int a, int b) {
33            int temp = 0;
34            while(b != 0){
35                temp = a % b;
36                a = b;
37                b = temp;
38            }
39            return a;
40        }
41    }
42 }
43

```