

1316: 【例4.6】数的计数(Noip2001)

题目描述

我们要求找出具有下列性质数的个数（包含输入的正整数 n ）。

先输入一个正整数 n ($n \leq 1000$)，然后对此正整数按照如下方法进行处理：

1. 不作任何处理；
2. 在它的左边拼接一个正整数，但该正整数不能超过原数，或者是上一个被拼接的数的一半；
3. 加上数后，继续按此规则进行处理，直到不能再加正整数为止。

输入格式

一行，一个正整数 n ($n \leq 1000$)。

输出格式

一个整数，表示具有该性质数的个数。

输入样例

6

输出样例

6

解析

以测试数据6为例，我们首先可以在它的左边加上1得到16，那么1的左边还能继续加数字吗？根据题意是不可以的，因此我们得到了1种方法，即 $f[1]=1$ 。

我们继续在6的左边加上数字2，得到了26，仔细观察，我们发现还能继续添加1，于是就得到了126，则有 $f[2] = f[1]+1=2$ (数字2本身)。

同理，我们在6的左边加上3，就能得到36，而在3的左侧我们还能加入1，则有 $f[3]=f[1]+1=2$ (数字3本身)。

最终，我们发现 $f[6]=f[1]+f[2]+f[3]+1=1+2+2+1=6$ ，整理成递推公式则有：

$$f[n] = f[n/2]+f[n/2-1]+\cdots+f[1]+1$$

很明显 $f[1] = 1$ 。本题注意由于数据量较大，需要使用缓存。

编码

```
#include <bits/stdc++.h>

using namespace std;

//缓存数组
int f[1010];

int Count(int n) {
    //判断缓存中是否存在数据
    if (f[n] != -1) {
        return f[n];
    }
    //开始调用递推公式
    int ans = 1;
    for (int i = 1; i <= n / 2; i++) {
        ans += Count(i);
    }
    //将数据存入缓存
    return f[n] = ans;
}

int main(int argc, char **argv) {
    //将数组初始化为-1代表未被赋过值
    memset(f, -1, sizeof(f));
    int n;
    //初始数据为1
    f[1] = 1;
    cin >> n;
    cout << Count(n);
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

