

01-A-2 算法有穷性

#数据结构邓神

```
序列 Hailstone(n) =  
{  
  {1} n<=1  
  {n} 并上 Hailstone(n/2) n 是偶数  
  {n} 并上 Hailstone(3*n + 1) n 是奇数  
}
```

对于任何一个自然数 n 都可以定一个 hailstone 序列

比如开始 $n = 10$

10 是偶数 {10,5}

5 是奇数 {10,5,16}

16 是偶数 {10,5,16,8}

8 是偶数 {10,5,16,8,4}

4 是偶数 {10,5,16,8,4,2}

2 是偶数 {10,5,16,8,4,2,1}

1 是奇数 结束

我们发现这个序列有时候会上升，有时候会上升

他有时候会持续的下降，但是不会持续的上升 ($3*n+1$ 必然是偶数)

程序设计：

```
// 计算 hailstone 长度  
int hailstone(int n){  
    int length = 1;  
    while(1<n){  
        (n % 2) ? n = 3 * n + 1 : n /= 2;  
        length++;  
    }  
    return length;  
}
```

我们要如何证明这个函数是有穷的，就是对任何的输入都能经过有限次数的步骤后退出，那么这个while循环能做到吗？

这取决于这个序列的定义

对于任意的 n 总有 $|Hailstone(n)| < \infty$???

这个问题暂时没有结论!

所以对于这个函数未必是一个算法, 他不确定是否有穷