

```
/*
泰波那契序列  $T_n$  定义如下：

 $T_0 = 0, T_1 = 1, T_2 = 1$ ，且在  $n \geq 0$  的条件下  $T_{n+3} = T_n + T_{n+1} + T_{n+2}$ 

给你整数  $n$ ，请返回第  $n$  个泰波那契数  $T_n$  的值。
```

示例 1：

输入： $n = 4$   
输出：4  
解释：  
 $T_3 = 0 + 1 + 1 = 2$   
 $T_4 = 1 + 1 + 2 = 4$

示例 2：

输入： $n = 25$   
输出：1389537

提示：

$0 \leq n \leq 37$   
答案保证是一个 32 位整数，即  $\text{answer} \leq 2^{31} - 1$ 。

来源：力扣 (LeetCode)

链接：<https://leetcode-cn.com/problems/n-th-tribonacci-number>

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

```
*/
```

分析：

- 类似斐波那契数列，常规的方法是迭代法和递归法。
- 递归法很遗憾的超时了。

方法一:C++,迭代法

```
class Solution
{
public:
    int tribonacci(int n)
    {
        int t[3] = {0,1,1} ;
        int i = 0 ;
```

```

        int temp    =    0        ;

        if(n<0)
        {
            return 1;
        }
        else if(n<=2)
        {
            return t[n];
        }
        else
        {
            while(n>=3)
            {
                temp = t[0] + t[1] + t[2];
                t[0] = t[1];
                t[1] = t[2];
                t[2] = temp;
                n --;
            }
            return t[2];
        }
    }
};

```

/\*

执行结果：通过

[显示详情](#)

执行用时 : 4 ms, 在所有 C++ 提交中击败了100.00%的用户

内存消耗 : 8.1 MB, 在所有 C++ 提交中击败了100.00%的用户

\*/

方法二:C++,递归法

```

class Solution
{
public:
    int tribonacci(int n)
    {
        if(n<=0)
        {
            return 0;
        }
        if(n==1)
        {
            return 1;
        }
        if(n==2)
        {
            return 1;
        }
        return tribonacci(n-1) + tribonacci(n-2) + tribonacci(n-3);
    }
};

```

```
}  
};
```

```
/*
```

执行结果：

超出时间限制

显示详情

最后执行的输入：

33

```
*/
```

---

AlimyBreak

2019.07.31