

LeetCode第1137号问题：第N个泰波那契数

本文首发于公众号「图解面试算法」，是 [图解 LeetCode](#) 系列文章之一。

个人博客: www.zhangxiaoshuai.fun

本题选自leetcode中第1137题，easy级别，目前通过率52.4%

题目描述：

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 $\text{answer} \leq 2^{31} - 1$ 。

题目分析：

要是之前有接触过斐波那契数列的话，这道题是很容易有解决思路的。我们有以下三种方法（正经方法两种，哈哈）来解决该问题：

1. 递归（但是leetcode中是无法AC的，超出时间限制，但是还是会将代码展示出来）
2. 动态规划（这种题都是已知前面的来求得未知的，使用dp再合适不过）
3. 暴力（抖机灵，看一乐就可以啦）

GIF动画演示：

代码：

递归版本：

```
public int tribonacci(int n) {  
    if (n == 0) {  
        return 0;  
    }  
}
```

```

    if (n == 1 || n == 2) {
        return 1;
    }
    return tribonacci(n - 1) + tribonacci(n - 2) + tribonacci(n - 3);
}

```

动态规划

```

int[] dp = new int[38];
public int tribonacci(int n) {
    if (dp[n] != 0) {
        return dp[n];
    }
    if (n == 0) {
        return 0;
    } else if (n == 1 || n == 2) {
        return 1;
    } else {
        int res = tribonacci(n - 1) + tribonacci(n - 2) + tribonacci(n - 3);
        dp[n] = res;
        return res;
    }
}

```

暴力法（十分暴力，哈哈哈哈.....）

```

public int tribonacci(int n) {
    int[] Ts = {0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, 3136,
5768, 10609, 19513, 35890, 66012, 121415, 223317, 410744,
755476, 1389537, 2555757, 4700770, 8646064,
15902591, 29249425, 53798080, 98950096,
181997601, 334745777, 615693474, 1132436852, 2082876103};
    return Ts[n];
}

```