10 博弈型动规

笔记本： DP Note
创建时间： 10/13/2019 3:16 PM **更新时间：** 10/31/2019 10:37 PM
作者： tanziqi1756@outlook.com

394. Coins in a Line

中文 English

There are `n` coins in a line. Two players take turns to take one or two coins from right side until there are no more coins left. The player who take the last coin wins.

Could you please decide the **first** player will win or lose?

If the first player wins, return `true`, otherwise return `false`.

Example
Example 1:

```
Input: 1
Output: true
```

Example 2:

```
Input: 4
Output: true
Explanation:
The first player takes 1 coin at first. Then there are 3 coins left.
Whether the second player takes 1 coin or two, then the first player can t
ake all coin(s) left.
```

Challenge
O(n) time and O(1) memory

```java
public class Solution {
    /**
     * @param n: An integer
     * @return: A boolean which equals to true if the first player will win
     */
    public boolean firstWillWin(int n) {
        // write your code here
        return n % 3 != 0;
    }
}
```

Accepted Powered by LintCode FlashJudge

100%

100% test cases passed Total runtime 172 ms

用普通的动规

动态规划组成部分二：转移方程

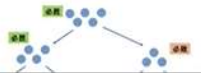
九章算法

- 设f[i]表示面对i个石子，是否先手必胜(f[i] = TRUE / FALSE)

f[i] =

TRUE，f[i-1]==FALSE AND f[i-2]==FALSE	拿1或2个石子都必胜
TRUE，f[i-1]==FALSE AND f[i-2]==TRUE	拿1个石子必胜
TRUE，f[i-1]==TRUE AND f[i-2]==FALSE	拿2个石子必胜
FALSE，f[i-1]==TRUE AND f[i-2]==TRUE	必败

f[i] = f[i-1] == FALSE OR f[i-2] == FALSE



```java
public class Solution {
    /**
     * @param n: An integer
     * @return: A boolean which equals to true if the first player will win
     */
    public boolean firstWillWin(int n) {
        // write your code here
        if( n == 0 ) {
            return false;
        }

        boolean[] dp = new boolean[n+1];
        dp[0] = false;
        dp[1] = true;

        // State: dp[i]: 面对i个石头时，必胜还是必败
        for( int i = 2; i < n+1; i++ ) {
            // 拿走1个或两个石头，让对方必败时，这种状态就为true
            dp[i] = (dp[i-1] == false || dp[i-2] == false );
        }

        return dp[n];
    }
}
```

必做

多选题 **下面是必胜状态的是?**

A 没有后继状态的状态是必败状态

B 当存在至少一个必败状态为它的后继状态

C 当它的所有后继状态均为必胜状态

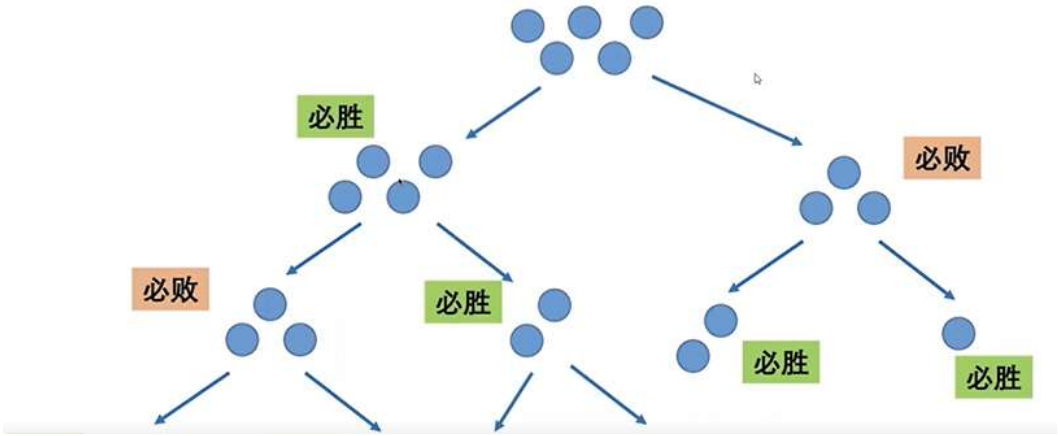
D 胜利的概率大于失败的概率

已回答

答错了！正确答案是 A B，有19%的同学答对了，加油赶上他们!

解析：C的状态为必败状态，对于B来说如果该状态至少有一个后继状态为必败状态，那么玩家可以通过操作到该必败状态；此时对手的状态为必败状态——对手必定是失败的，而相反地，自己就获得了胜利。

博弈动态规划：必胜 vs 必败



必做

多选题 **有 n 个不同价值的硬币排成一条线，第i个硬币的价值是value[i]。两个参赛者轮流从左边依次拿走 1 或 2 个硬币，直到没有硬币为止。计算两个人分别拿到的硬币总价值，价值高的人获胜。请问是先手获胜还是后手获胜？这个问题的子问题是？**

A 拿了编号为n-1的硬币，剩下n-1个硬币时先手获得的最大价值是多少？

B 拿了编号为0的硬币，剩下n-1个硬币时先手获得的最大价值是多少？

已回答

没有答对哦！正确答案是 B ，有41%的同学做对了这道题目哦，继续努力！

解析：题目中是从左往右的。所以比他的子问题应该是拿了左边的硬币，也就是编号为0的硬币。

多选题 **如果按照上一题的子问题求解，那么对应的状态转移方程是什么？（sum[i]表示拿走前i个硬币时总价值，f[i]表示拿走前i个硬币时先手获得的最大价值）**

A

```
sum[i] = sum[i+1] + values[i];
f[i] = max(
        sum[i+1] - f[i+1] + values[i],//拿一个硬币
        sum[i+2] - f[i+2] + values[i] + values[i+1]//拿两个硬币
);
```

B

```
sum[i] = sum[i-1] + values[i];
f[i] = max(
        sum[i-1] - f[i-1] + values[i],//拿一个硬币
        sum[i-2] - f[i-2] + values[i] + values[i-1]//拿两个硬币
);
```

C

```
f[i] = max(
    value[i] + f[i + 1],//拿一个硬币
    value[i] + value[i + 1] + f[i + 2]//拿两个硬币
);
```

D

```
f[i] = max(
    value[i] + f[i - 1],//拿一个硬币
    value[i] + value[i - 1] + f[i - 2]//拿两个硬币
);
```

正确答案是 A，有11%的同学答对了，要加油了。

解析：我们拿走第i个硬币的最大价值就应该是拿到i的时候硬币总价值减去对手拿的在加上我们自己拿的硬币的价值。并且我们根据上一题的子问题来遍历的话最后一个拿的硬币是0号硬币，然后是1号硬币，所以需要我们从后往前遍历。

单选题 **上题中的初始态正确的是什么?**

```
int[] f = new int[n+1];
```

A
```
f[n+1] = sum[n+1] =
0;
f[n] = sum[n] = 0;
```

B
```
f[n] = sum[n] = 0;
f[n-1] = sum[n-1] = v
alues[n-1];
```

C
```
f[n-1] = sum[n-1] =
0;
f[n-2] = sum[n-2] = v
alues[n-1];
```

我不会

正确答案是 B，有52%的同学做对了这道题目哦，继续努力!

解析：AB都是对的初始化，但是题目中f数组的大小为n+1，数组下标是从0-n的，所以不存在f[n+1]。
C选项正确的形式应该是f[n-1] = sum[n-1] = values[n-1]; f[n-2] = sum[n-2] = values[n-1] + values[n-2];的，不让在状态转移的时候会缺少一项

单选题 **如果我们选择上题中B选项作为初始态，我们的循环顺序应该是怎么样的?**

A 从n到0

B 从n-1到0

C 从n-2到0

D 从n-3到0

我不会

正确答案是 C，有36%的同学超过了你，但是千万不要气馁。

解析：我们初始化定于的数组长度是n+1的，而状态转移方程中有i+2，所以选择AB选项会导致数组越界，而从n-3开始的话，我们就没有对n-2进行赋值，导致后续的递推出错。

多选题 **结合上面四个题目的选项和答案，我们最后需要怎么判断先手获胜呢?**

A f[0] > sum[0] / 2

B f[0] < sum[0] / 2

C f[0] > sum[0] - f[0]

D f[0] < sum[0] - f[0]

我不会

正确答案是 A C ，有24%的同学做对了这道题目哦，继续努力！

解析：1.前面我们说到sum为硬币的总价值，f为先手获得的最大价值，那么后手获得的最大价值就是sum[0] - f[0]，我们要判断谁赢只需要根据题目意思比较一下先手和后手谁获得价值大就行。
2.sum[0] / 2表示总价值的二分之一，如果先手获得的价值已经大于一半了，那么肯定是先手获胜