# Tiling with Dominoes

[SPOJ: M3TILE solution explanation](https://stackoverflow.com/questions/16388579/spoj-m3tile-solution-explanation#)
https://stackoverflow.com/questions/16388579/spoj-m3tile-solution-explanation#

首先定义T(n)为 3xn的放置方案数，P(n)为3xn（缺少一个角）的放置方案（n ≥ 4）

**对于T(n)，假设从左上角开始放置，可以分两种情况：**

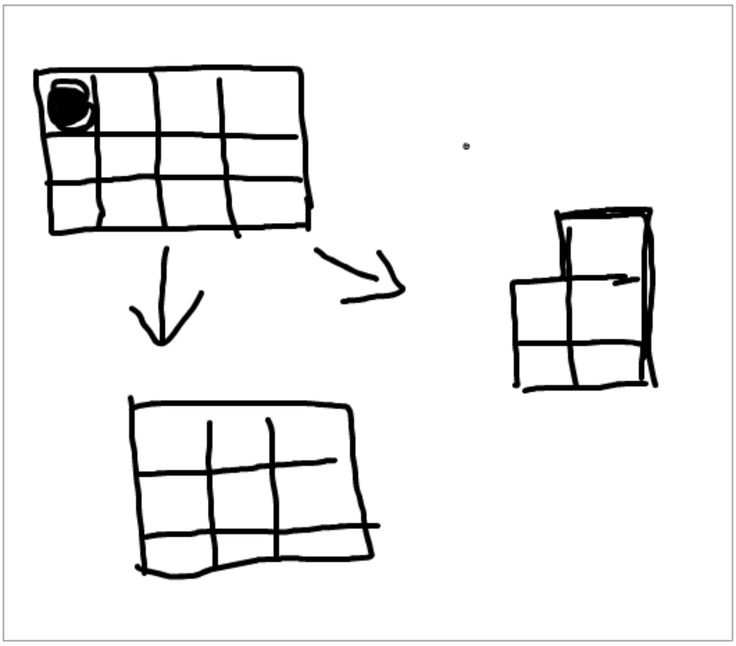
1. 左上角放了一个2×1的，那么左下角必然放置一个1×2的，那么问题转换求P(n-1)
2. 左上角放了一个1×2的，那么左下角又分为两种情况：
    1. 左下角放1×2的，此时中间必然也放1×2的，那么问题转为求T(n-2)
    2. 左下角放2×1的，那么问题又转换为求P(n-1)

所以，有：

$$T(n) = 2 * P(n-1) + T(n-2) \quad (1)$$

接下来考虑P(n-1)（以3×4为例子并假设左上角缺失）：

P(n-1)有如下两种状态转移方案



所以，有：

$$P(n-1)=T(n-2)+p(n-3)$$

将上式带入（1）中，有：

$$T(n)=3*T(n-2)+P(n-3) \quad (2)$$

再令（1）中n为n-2，则有：

$$T(n-2)=3*P(n-2)+T(n-4)$$

将上式带入（2）中，得：

$$T(n)=4*T(n-2)-T(n-4)$$

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

using namespace std;

// dp[i] 表示 3xi 的方案数
int waysOfDominoes(int n) {
    vector<int> dp(n + 1);
    dp[0] = 1, dp[1] = 0, dp[2] = 3, dp[3] = 0;
    for (int i = 4; i <= n; ++i)
```

```
        dp[i] = 4 * dp[i - 2] - dp[i - 4];
    return dp[n];
}

int main() {
    int n = 12;
    printf("%d\n", waysOfDominoes(n));

}
```