赵州桥

试题解析：

这道题目是一道数学题，它由两个部分组成的。其一是求染色方案数，其二是对加权后的方案数的求和。

首先我们要解决第一个问题，求出用 m 种颜色对 2×n 的桥的染色的总方案数。

我们用 f[i]表示用 m 种颜色对桥的前 i 个单位长度的染色方案数，(i,1)表示桥上第 i 个单位的上面一个格子，(i,2)表示桥上第 i 个单位的下面一个格子。

现在我们对桥上第 i + 1 个单位上的两个格子进行染色。如果(i + 1,1)上的颜色和(i,2)的颜色相同，则(i + 1,2)的颜色有(m – 1)种情况;如果(i + 1,1)上的颜色和(i,2)的颜色不相同，则(i + 1,2)的颜色有(m – 2)种情况。

有 $f[i + 1] = f[i] \times (m - 2)^{2} + f[i] \times (m - 1) = f[i] \times (m^2 - 3 \times m + 3)$。又因为 $f[1] = (m - 1) \times m$，所以 $f[n] = f[1] \times (m^2 - 3 \times m + 3)^{n-1}$。

可以得到题目中所求的值即为 $f[i] \times (2\ i)^m$。

之后的过程就直接决定这道题目的得分了。

方法一：暴力

对于前 25%的数据，因为 $n \le 10^6$，直接暴力循环计算出每个 $f[i] \times (2\ i)^m$，其中$(2\ i)^m$用快速幂。这样时间复杂度为 O(nlog(m))。

方法二：矩阵乘法

通过一定的计算我们可以发现：

$f[i + 1] \times (2\ i + 2)^{m} = f[i + 1] \times C(m, k) \times (2\ i)^{k} \times 2^{m-k}$

又因为 $f[i + 1] = f[i] \times (m^{2} - 3 \times m + 3)$。

因此我们可以另外设一个函数 $g[i, k] = f[i] \times (2\ i)^{k}$

由上述的推算我们可以发现：

$g[i + 1, k] = g[i, j] \times (m^{2} - 3 \times m + 3) \times C(k, j) \times 2^{k-j}$

这样我们就可以把 g 这个函数做成一个 $1 \times m$ 的行向量，进行矩阵乘法求得每一个 g[i, m]的值。但是题目要我们求的是 g[i, m]的总合，这样我们需要在矩阵上面多加一个元素，不妨将其设做 S[i]，且 S[i] = g[k, m]。这样我们在计算矩阵的时候把 S 跟在 g 的后面形成一个 $1 \times (m + 1)$的行向量，且每次转移的时候把 g[i, m]和 S[i]加在 S[i + 1]上面。

这样时间复杂度为 $O(m^{3}\log(n))$的，可以过掉 65%的数据。

方法三：根据矩阵特性的优化

根据上述的转移方程，我们可以发现其实这个转移状态的矩阵始终是一个上三角矩阵。因此我们可以把矩阵乘法的 3 个循环改写成：

```
for (int i = 0; i ≤ m + 1; i ++)
    for (int j = i; j ≤ m + 1; j ++)
        for (int k = i; k ≤ j; k ++)
```

这样就可以再多过 15%的数据。

其实这就是这道题目的正解了，但是后 20%的数据特别奇葩，m 都高达 3000。

显然，这是不可能靠优化来解决的。。。。。。

再观察数据范围会发现 p 特别的小，而对 $f[i+1] \times (2i+2)^m$，$f[i] \% p$、$(2i+2) \% p$ 会有小于 p 的循环节，故 $f[i+1] \times (2i+2)^m \% p$ 会有小于 p^2 的循环节，所以我们只要找循环节就可以解决最后这 20%的数据。