知识精炼（一）

主讲人：邓哲也

POJ 1681 Painter’ s Problem

有一个 n×n 的方格棋盘。有的格子是白色的，有的格子是黄色的。 Bob 的粉刷(i, j)的时候，(i, j)，（i-1, j），(i+1, j), (i, j-1), (i, j+1) 颜色都会发生改变。

现在请你计算 Bob 最少粉刷几个格子就可以使整个棋盘都变成黄色呢？

$n \leqslant 15$

POJ 1681 Painter’ s Problem

每次粉刷一个格子，格子四周的四个格子都会发生变化。

我们用一个数组 A[x][y] 来表示是否粉刷格子(x,y)。

如果 A[x][y] = 1，就粉刷；A[x][y] = 0，不粉刷。

POJ 1681 Painter’s Problem

同理，一个格子的状态是受它四周的四个格子是否被粉刷决定的。

检查 A[x - 1][y], A[x + 1][y], A[x][y + 1], A[x][y - 1] 和 A[x][y] 本身。

如果 5 个数的和是奇数，就说明(x, y) 格子的状态发生了改变。

等价于计算这 5 个数的异或和。

POJ 1681 Painter’ s Problem

令 B[x][y] 表示格子(x,y) 的初始状态。

白色则 B[x][y] = 1, 黄色则 B[x][y] = 0

最后的目标是 B[x][y] 变成 0.

也就是 B[x][y] xor A[x-1][y] xor A[x+1][y] xor A[x][y-1] xor A[x][y+1] xor A[x][y]= 0

对于每个格子我们都可以列出一个异或方程。

POJ 1681 Painter’s Problem

总共可以得到 n^2 个方程。

解出这个方程，在所有的解中找到 Sum(A[1..n][1..n]) 最小的一组解。A 数组的和就是答案。

使用高斯消元法，时间复杂度在 $O(n^6)$

POJ 1681 Painter’s Problem

注意高斯消元的过程中，可能出现无解或无穷解。

无解：输出 inf 即可。

无穷解：说明有自由元。为了让操作数量越少越好，把它视作 0 即可。

POJ 1681 Painter’s Problem

```
int dx = {0, 0, -1, 0, 1};
int dy = {0, 1, 0, -1, 0};
int a[250][250];

void init(){
    for (int i = 0;i < n;i ++)
        for(int j = 0;j < n;j ++)
            for(int k = 0;k < 5;k ++){
                int xx = i + dx[k], yy = j + dy[k];
                if (xx >= 0 && yy >= 0 && xx < n && yy < n)
                    a[i * n + j][xx * n + yy] = 1;
            }
}
```

POJ 1681 Painter’s Problem

```
int guass(){
    for (i = j = 0; i < n * n && j < n * n;i ++, j ++){
        k = i;
        for (k = i;k < n * n;k ++)
            if (a[k][j]) break;
        for (int l = j;l <= n * n;l ++)
            swap(a[i][l], a[k][l]);
        if (!a[i][j]){
            i --;
            continue;
        }
        for(k = i + 1;k < n * n;k ++)
            if (a[k][j])
                for(int l = j;l <= n * n;l ++)
                    a[k][l] ^= a[i][l];
}
```

POJ 1681 Painter’s Problem

```
    k = i;
    for (i = k;i < n * n;i ++)
        if (a[i][n * n])
            return 0;
    for (i = k - 1;i >= 0;i --)
        for (j = i + 1;j < n * n;j ++)
            a[i][n * n] ^= (a[i][j] & a[j][n * n]);
    return 1;
}
```

HDU 5833 Zhu and 772002

有 n 个数，A[1]，A[2]，⋯ ，A[n]。

其中每个数的质因子都不超过 2000。

你可以选出几个数，使得他们的乘积是完全平方数。

问几种方案。

$n \leqslant 300$，$1 \leqslant A[i] \leqslant 10^{18}$

样例：3 3 4 （方案数：3）

样例：2 2 2 （方案数：3）

HDU 5833 Zhu and 772002

首先注意一点，题中的数的质因子都不超过 2000.

不超过 2000 的质数只有 303 个。

因此，选出的数乘出来的结果里的不同的质因子最多也只有这 303 个。

完全平方数满足：对于每个质因子，它的幂次一定是偶数。

HDU 5833 Zhu and 772002

因此我们可以用 X[i] 来表示是否选了第 i 个数。

B[i][j] 表示第 i 个质数在 A[j] 中出现的次数。

那么要满足选出的是完全平方数，也就是第 i 个质数出现的次数被2整除。

$$B_{i,1}X_1 + B_{i,2}X_2 + \cdots + B_{i,n}X_n = 0 \pmod 2$$

注意到模 2 的加法就是异或。

所以可以把系数 B 先对 2 取模。

然后得到：$B_{i,1}X_1 \text{ xor } B_{i,2}X_2 \text{ xor } \cdots \text{ xor } B_{i,n}X_n = 0$

HDU 5833 Zhu and 772002

因此这样就是 n 个未知数，303 个方程的异或方程组。

用高斯消元法即可解出。

统计自由元的个数 k。

那么对于这些自由元，我们有 2^k 种方案。

减去全都不选的方案，答案就是 $2^k - 1$ 。

下节课再见