知识精炼（二）

主讲人：邓哲也

POJ 3150 Cellular Automaton

有 n 个数排成一个环。

定义一次变换为：把这个数变成距离它不超过 d 的位置上的数之和对 m 取模的值。

问这样变换 k 次后，每个位置上的数是多少。

n <= 500, m <= 10^9, k <= 10^9

POJ 3150 Cellular Automaton

对于每一次变换，其实可以写成一个矩阵：

比如题中 d = 2

$$
矩阵\ A = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}
$$

POJ 3150 Cellular Automaton

但是这么做时间复杂度是 $O(n^3 \log k)$

不能通过。

我们可以看看 A 的幂次长什么样

$$A^2 = \begin{bmatrix} 3 & 2 & 1 & 1 & 2 \\ 2 & 3 & 2 & 1 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 1 & 2 & 3 & 2 \\ 2 & 1 & 1 & 2 & 3 \end{bmatrix} \quad A^3 = \begin{bmatrix} 7 & 6 & 4 & 4 & 6 \\ 6 & 7 & 6 & 4 & 4 \\ 4 & 6 & 7 & 6 & 4 \\ 4 & 4 & 6 & 7 & 6 \\ 6 & 4 & 4 & 6 & 7 \end{bmatrix}$$

POJ 3150 Cellular Automaton

$$A^2 = \begin{bmatrix} 3 & 2 & 1 & 1 & 2 \\ 2 & 3 & 2 & 1 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 1 & 2 & 3 & 2 \\ 2 & 1 & 1 & 2 & 3 \end{bmatrix} \quad A^3 = \begin{bmatrix} 7 & 6 & 4 & 4 & 6 \\ 6 & 7 & 6 & 4 & 4 \\ 4 & 6 & 7 & 6 & 4 \\ 4 & 4 & 6 & 7 & 6 \\ 6 & 4 & 4 & 6 & 7 \end{bmatrix}$$

可以发现 A 的幂次都满足 A[i][j] = A[i-1][j-1]

POJ 3150 Cellular Automaton

因为如果 A 和 B 都满足 A[i][j] = A[i-1][j-1], B[i][j] = B[i-1][j-1]

那么 C[i][j] = sum(A[i][k] * B[k][j])

= sum(A[i-1][k-1] * B[k-1][j-1])

= sum(A[i-1][k] * B[k][j-1])

= C[i-1][j-1]

POJ 3150 Cellular Automaton

因此一个矩阵只要存下第一行就可以得到整个矩阵。

计算矩阵乘法的时候，只要枚举 n^2 次。

时间复杂度变成了 $O(n^2 \log k)$

POJ 3150 Cellular Automaton

```
void mul(int a[], int b[]){
        int c[505];
        memset(c, 0, sizeof(c));
        for (int i = 0;i < n;i ++)
                for(int j = 0;j < n;j ++)
                        c[i] = (1LL * a[j] * b[(i - j + n) % n] + c[i]) % m;
        return c;
}
```

下节课再见