

壮压DP

状压 DP 是动态规划的一种，通过**将状态压缩为整数**来达到优化转移的目的。

——OI-Wiki

常见的位运算

$(n \gg k) \& 1$

$n \& (1 \ll (k - 1))$

$n \text{ xor } (1 \ll k)$

$n | (1 \ll k)$

成对变换: $x \text{ xor } 1$

`__builtin_`

[P1896 \[SCOI2005\] 互不侵犯](#)

考虑到此题计数不重复需要以一整行为状态进行DP，所以将长度为 n 的一行压缩为一个长度为 n 的二进制数，

其中第 i 位为 1/0 代表这行第 i 列是否放国王

所以可以以 $f_{i,j,k}$ 表示第 i 行的状态为 j ，用了 k 个国王的情况下的方案数

$$f_{i,j,k} = \sum_{j^1} f_{i-1,j^1,k-\text{cnt}_j}$$

(转移条件省略了kkk)

注意需要预处理出相邻位不同时为 1 的状态以及这些状态中 1 的个数

[P2704 \[NOI2001\] 炮兵阵地](#)

先考虑与上题相似的状态设置，

故用 $f_{i,j,k}$ 表示第 i 行的状态为 j ，第 $i - 1$ 的状态为 k 的时候能放下的最多炮兵

$$f_{i,j,k} = \max_l f_{i-1,k,l}$$

此外，本题的状态还可以设成另一种形式，考虑到行之间转移的时候只需要考虑每一列的上面和上面的上面是否有炮兵即可，

所以可以用三进制数表示一行的状态，第 i 位为 2/1/0 表示第 i 位有炮兵/上方有炮兵/没有炮兵

[P2157 \[SDOI2009\] 学校食堂](#)

观察到 B_i 只有 7,

令 $f_{i,j,k}$ 表示前 $i-1$ 位同学都已经拿到饭, 第 i 及后 7 位同学是否拿饭的状态为 j , 当前最后拿到饭的同学为 $i+k$ 所需的最短时间,

容易发现当第 i 位同学拿到饭的情况下, 有转移:

$$f_{i+1,j>>1,k-1} = f_{i,j,k} \quad (j \& 1 = 1)$$

否则可以枚举第 $i+h$ 位同学中有人拿到了饭, 有转移:

$$f_{i,j|(1<<h),h} = f_{i,j,k} + Time(i+k, i+h)$$

注意判断其中有没有同学不能容忍

[P1357 花园](#)

容易得到:

$$f_{i,j} = f_{i-1,j>>1} + f_{i-1,(j>>1)|(1<<m)}$$

用矩阵加速即可