1. **种植**(plant.c/cpp/pas)

**前100％的数据**：

非常裸的状压DP，大家应该都会做，PPT上面也讲过差不多的题（放置国王），这里就不讲了，具体可以参考代码。

1. **计数**(count.c/cpp/pas)

**前100％的数据**：

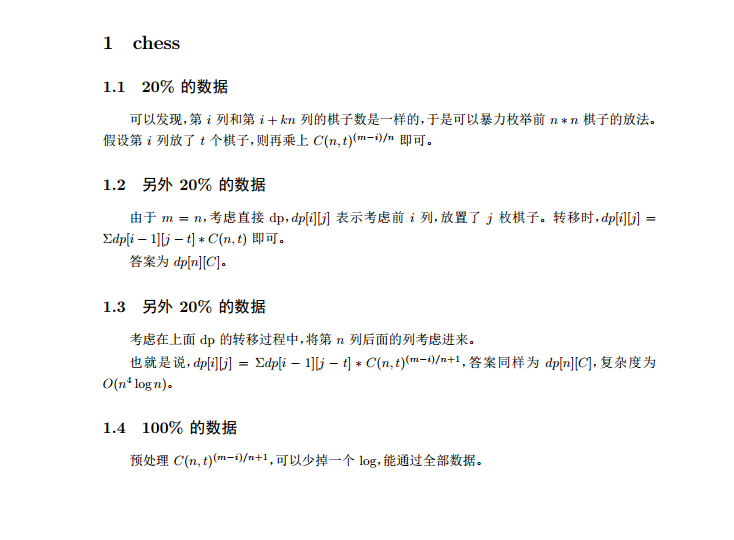
转化成求[0,R]上各位数字之和能整除原数的数个数。

整除的一般套路是在状态中记录对模数取模的值，但是发现在dp过程中并不能确定最后的各位数字之和是多少，所以模数难以确定。

但是发现各位数字之和至多9\*18，非常地小。所以考虑枚举模数mod，在dp过程中记录对mod取模的值，即dp[i][j][k][0/1]表示处理完前i位，各位和是j，对mod取模的值是k，是否压上界的方案数。最终统计答案时仅将 dp[num][mod][0][0/1] 统计到答案中，其中num是R的位数。

3 **棋盘**(chess.c/cpp/pas)

这道题题解之前用LaTeX写过，这里直接用截图方便一些。



4 **树**(tree.c/cpp/pas)

**100% 的数据**：

注意到一个性质：假设存在一个大小为k的联通块，其中含有 a 个黑点，存在另一个大小为 k 的联通块，其中含有 b 个黑点。那么对于任意 a<=i<=b，一定都存在一个大小为k的联通块，其中含有 i 个黑点。证明：对于含有a个黑点的联通块，我们每次从联通块中删去一个点，再加入一个新的点，形成一个新的大小为k的联通块，那么黑点个数至多改变1；而通过这样的过程我们能够从含有a个黑点的联通块逐渐变成含有b个黑点的联通块，所以a到b中任意一个数均会出现。

知道这个性质以后，就是一个简单的树上背包。f[i][j]表示以i为根，大小为j的联通块（根一定在联通块中）黑点数的最小值；g[i][j]表示以i为根，大小为j的联通块（根一定在联通块中）黑点数的最大值。

那么即f[a][j] = min(f[a][j], f[a][j-k]+f[b][k]) ，g[a][j] = max(g[a][j], g[a][j-k]+g[b][k])。

枚举的时候注意 j 从大到小枚举（应该类似于01背包而不是完全背包），然后枚举k的时候要注意上下界，确定上下界的准则就是考虑上枚举的黑点数一定不会超过结点总数。具体可以参考标程。

计算完毕之后，可以再用minn[i]和maxn[i]分别表示整棵树中，大小为i的联通块，含有的黑点的最少数目和最大数目。

于是处理询问的时候，对于询问x, y，即询问是否存在大小为 x 的联通块，包含 y 个黑点。即判断 y 是否大于等于minn[x]并且小于等于maxn[x]即可。