基础思路：DP 套 DP

显然能够发现如果设 $f_{i,0/1}$ 表示合法麻将牌集合 S 的大小为 i，且这些牌的花色相同，其中包不包含一个对子的数量，则在求出 $f_{i,0/1}$ 的基础上，能用组合数求出答案。

具体而言，设 $g_{i,j}$ 表示共有 i 种花色，集合大小为 j 的方案数。那么可以用 f 背包求出 g，然后再用 g 组合数即可。因为对于每种花色，f 中可用于背包的项共 $O(3k)$ 个，每次对 f 中的一项进行背包需要 $O(3k)$ 的复杂度，不同的花色共 $O(k)$ 种，所以通过 f 计算 g 的复杂度是 $O(k^3)$。

考虑如何求出 $f_{i,0/1}$，我们能够设一个 dp，然后从 1 到 n 依次考虑每个大小的牌加多少个，现在

需要判断的是加进去之后是否合法。

转化为一个这样的命题：给定一个集合，能否胡牌。有一个简单的 dp，设 $dp_{i,s_1,s_2,0/1}=[0,1]$ 表示当前考虑第 i 张牌，从 $i-2$ 开始的顺子有 s_1 个，从 $i-1$ 开始的顺子有 s_2 个，有没有对子的情况是否合法。转移显然。最终如果 $dp_{\text{len}+1,0,0,1}=1$ 则说明该集合能胡牌，否则不行。

显然 $0\le s_1,s_2\le 4$，但是如果在一种胡牌方案中，出现了三个一样的顺子，那么可以转换为三个相邻的刻子，所以 $0\le s_1,s_2\le 2$，因此 $(s_1,s_2,0/1)$ 的总状态数量为 $3\times 3\times 2=18$，也就是说，用一个大小为 18 的数组就能描述上述 dp，从而表示当前麻将牌集合是否最终可以胡牌。

那么为了计算可以胡牌的集合的数量，我们设 $dp_{i,j,S}$ 为，当前加入到数字 i，一共加入了 j 张牌，胡牌 dp 状态的集合为 S 的方案数。转移显然。

不过 2^{18} 种方案显然是非常大的，我们能从初始状态预处理出能到达的所有状态，经打表一共只有 68 种。

68 种的预处理可以参考 ZJOI2019 麻将的 bfs 写法，接着套用上面那个 dp 就行了。该部分时间复杂度为 $O(68nk)$。

总复杂度为 $O(68nk+k^3)$。

思路优化

k 较小的情况下，发现有大量的数字都不会被加入。

那么我们只需考虑若干连续的数字组成的方案数，然后再合并即可。

考虑每一个能胡的麻将牌集合中同一种花色的所有牌，设其集合大小为 s，显然 $s=3n+2$ 或者 $s=3n$，且能够将其分为若干段，这若干段每段的数字连续（可重复），但是不同段之间的数字不连续，显然对每段的牌内部也合法。那么设 $f_{i,j}$ 表示分出来的某一段数字值域长度为 i，所用牌数为 j 的方案数，这个可以用前面基础思路中的方法计算，然后再令 $g_{i,j,k}$ 表示同一种花色的牌可被分为 i 个不相接的段，共有 j 个不同的数字，所用牌数为 k 的方案数，这个可以用求得的 f 背包得到。

注意这个 $g_{i,j,k}$ 不考虑具体的牌的大小，但是考虑不同段之间大小的相对顺序。

- 例如对于牌组 $\{2,3,4,6,7,8,9,9\}$ 在 $g_{i,j,k}$ 中与 $\{1,2,3,6,7,8,9,9\}$ 相同，但是与 $\{2,3,3,5,5,7,8,9\}$ 不相同。

求得 $g_{i,j,k}$ 后可以用组合数求出考虑具体牌的大小后同一种花色用 i 张牌是否有刻子的方案数 $h_{i,0/1}$，也就是基础思路中的 $f_{i,0/1}$，之后用组合数背包 $O(k^3)$ 求出考虑多个花色的答案即可。

需要注意的是，虽然 f 的取值和 g 无关，但是如果我们先求出 f 再用这个固定的 f 去转移 g 的话，由于对于 g 的每个 i，每个可行的 f 需要用 $O(k^2)$ 的时间更新，而可行的 f 数量为 $O(k^2)$，所以复杂度为 $O(k^5)$。这里的关键是我们“知道”每一段的具体信息，但我们其实并不关心。如果我们采用单步运算的想法（类似二维矩阵插头 dp 里一格一格递推，而不是用一行一行递推），在递推 g 的过程中转移 f，即令 $dp_{i,j,k,S}$ 表示有 i 段，j 个不同的数字，所用牌数为 k，当前段内胡牌可行性 dp 状态数组为 S 的方案数。则当固定 i 时，转移的复杂度和 f 一致，为 $O(3k \times 3k \times 68 \times 4)$，总复杂度为 $O(9k^3 \times 68 \times 4 + k^3)$。

常数优化

我们发现被卡常了，想一下怎么优化常数。

对于每个状态 S，当前合法的麻将牌集合大小模 3 的余数一定相等。

发现常数主要集中在转移 f 的地方，然后发现对预处理的 68 种状态，满足某个状态的牌，一定满足数量大小模 3 的余数大小一定，于是可以用这个性质优化 3 的常数，再用 j 优化一下 k 的上下界并适当循环展开一下即可。

时间复杂度：$O\left(68k^3\right)$。