1 串

首先求出总的段数 tot，然后分类讨论。

修改相邻的两个：$O(n)$ 枚举修改的位置，然后在 tot 上面加加减减最后求最小值即可。

修改不相邻的两个：我们可以知道每个位置修改后最优能在总段数减掉多少个段，我们设其为 $a[i]$。然后由于要不相邻，所以我们对 $a[i]$ 做一个最大值前缀和 $mx[i]$。然后我们枚举我们修改的后面的位置 i，便可以用 $tot - mx[i-2] - a[i]$ 去更新答案。

2 数

首先将所有数从小到大排序，并称为 $a[1]...a[n]$。

我们不妨用 $f[i]$ 表示前 i 个数能表示的正整数数的上限，即前 i 个数能表示 $1 \sim f[i]$，无法表示 $f[i]+1$。

那么如果存在最小的 i, 使得 $f[i-1]+1 < a[i]$, 那么说明 $f[i-1]+1$ 这个数无论怎么样也凑不出，因为前 $i-1$ 个数凑不出，后面的数都比它大。则输出 $f[i-1]+1$ 即可。

那么如果对于当前 i，满足 $f[i-1]+1 >= a[i]$，则 $f[i] = f[i-1]+a[i]$。因为如果取 $a[i]$，再在前 $i-1$ 个数范围内任取，可以取到 $a[i] \sim a[i]+f[i-1]$；如果不取 $a[i]$，则可以取到 $1 \sim f[i-1]$。

总做法就如此递推过来，如果发现 $f[i-1]+1 < a[i]$ 则输出。如果不存在这样的 i，输出 $f[n]+1$。

3 图

一个非常基础的压位 dp。

状态：sta 是一个 $0 \sim 2^n - 1$ 的数，是一个 n 位的 0/1 压位状态。$f[sta]$ 表示由 sta 里面 1 组成的子图，最少需要删掉多少条边才能使这个子图是一个 DAG。

转移: 预处理出 $r[i]$，也是一个 n 位的 0/1 压位状态，表示有哪些点有一条边指向 i，表示 i 号点的入边端点集合。count 是一个函数计算一个数二进制含多少个 1，你可以也用数组预处理。

$$f[sta] = \min_{j \in sta}(f[sta \oplus (1 << j)] + count(sta \& r[j])).$$

这个转移的思路就是在于当前你最终的图一定是一个 DAG，那么一定会存在一个入度为 0 的点，那么我们枚举该点然后删掉由其他店指向该点以及自环进行转移即可。

4 睿爸三角

结论：第 i 行中不被 7 整除的数的个数为 $i-1$ 在 7 进制下的每一位 +1 后的乘积。（多找找规律你也能找到对不？）

首先我们知道杨辉三角是组合数，第 i 行第 j 列代表 $\binom{i-1}{j-1}$。引入 lucas 定理。

lucas 定理，求 $\binom{n}{m}$ 对素数 p 的取模，等于 n, m 在 p 进制下逐位求组合数的乘积。即 $C(n, m)\%p = C(n/p, m/p) * C(n\%p, m\%p)\%p$。

那么很简单，若要 $\binom{n}{m} \mod 7 \neq 0$，只要 m 在 7 进制下的每一位都比 n 小即可。那么每一位能取的方案就是当前那一位 $+1$，最后总共能取的方案就是每一位 $+1$ 后的乘积。

那么我们问题转化为：求 $0 - n - 1$ 中每一个数的每一位 $+1$ 后乘积和。我们发现乘积和适用分配律，所以我们可以用数位 dp 求解。具体做法是一个非常非常简单的数位 dp，在此不再赘述，详情参见 *srwudi* 代码。