状态设计比较有意思的一道区间dp。定义 $dp[l][r]$ 表示消除区间 $[l,r]$，**要求 $l-1$ 和 $r+1$ 位置不消去**，所需的最少发射球数。

初始化

单球初始化：对于每个位置 i, 如果 $a[i]$ 与左右相邻球颜色都不同，则 $dp[i][i]=2$（需额外2个球才能消除），否则初始化为极大值（表示无法单独消除）。

状态转移

按照区间dp的经典套路，考虑一个区间最后一次消除是什么样的，就能将区间分成几个互相独立的部分。由于状态中要求 $l-1$ 和 $r+1$ 不能消去，最后一次剩的球一定不能于这两个颜色相同。分以下几种情况讨论：

前两种情况是剩下的球在原序列中就连续，把区间分成了独立的两部分。

1. 剩下了单独一个球 $a[i]$。形如[...o...]，o表示剩下的球,.表示已经消掉的。

   则 $a[i]\neq a[l-1],a[i]\neq a[r+1]$，否则会粘在一起。

   还要求 $a[i]\neq a[i-1],a[i]\neq a[i+1]$，这个条件是转移方程中隐含了的。

   转移：$dp[l][r]=dp[l][i-1]+dp[i+1][r]+2$

2. 剩下两个本来就连着的球 $a[i]=a[i+1]$。形如[...oo...]

   要求同1，$a[i]\neq a[l-1],a[i]\neq a[r+1]$，下面几种情况同样，不再写。

   转移：$dp[l][r]=dp[l][i-1]+dp[i+2][r]+1$

下面几种情况是剩下的球在原序列中不连续的，会把区间分成独立的若干部分。

3. 剩下两个不连着的球。形如[...o...o...]

   要求颜色一样 $a[i]=a[j]$，转移：
   $dp[l][r]=dp[l][i-1]+dp[i+1][j-1]+dp[j+1][r]+1$

4. 剩下两个和一个球撞上，直接消掉。形如[...oo...o...]或[...o...o...o...]或[...o...oo...]等。

   可以先预处理一下，每个球先和左（或右）的一个连起来的最小代价。

   转移：

   $dp[l][r]=dp[l][i-1]+dp[i+1][j-1]+suf[j]$

   $dp[l][r]=pre[i]+dp[i+1][j-1]+dp[j+1][r]$

5. 剩下两个和两个球撞上。形如[...oo...oo...]或[...o..o..o..o...]等。

   这个情况下就一定是左边、右边先两两连起来，再最后撞上，就可以用中间两个球预处理的分别往左、右连上的最小代价加起来。

   转移：$pre[i]+dp[i+1][j-1]+suf[j]$

   预处理的情况是形如[...o...o或o...o...]

答案就是 $dp[1][n]$。

不预处理直接暴力枚举转移是 $O(n^4)$ 的，总复杂度 $O(n^6)$。

带预处理复杂度 $O(n^4)$。

观察转移可以发现，虽然实际操作需要有先后顺序，但是在算代价的时候与顺序无关，只是把各个区间代价加起来。

所以可以记 $f[l][r][k]$ 表示从 l 开始到 r 合并了 k 个，分隔开的区间dp值之和。

转移 $f[l][r][k] = f[l][i][k-1] + dp[i+1][r-1]$

$dp[l][r] = f[l][i][k] + dp[i+1][r] + max(0, 3-k)$

复杂度 $O(n^3)$。