回文问题

下面我们把数字序列 S 看作字符串。

对于两个长度相等字符串 A, B，如果对于任意的 $1 \le l \le r \le |A|$，$A[l..r]$ 是回文串当且仅当 $B[l..r]$ 是回文串，称 A 和 B 回文等价。

S^R 表示把字符串 S 翻转得到的字符串。一个字符串 S 是回文串等价于 $S = S^R$。

由于某些定义可能会在后面使用，建议按顺序阅读。

2.1 算法一

不难发现 $c = 2$ 时答案一定为 2，直接输出 2 即可通过 Subtask 1，获得 10 分。

2.2 算法二

定义一个 S 的子串 $S[l..r]$ 是极长回文子串，当且仅当它是一个不可向两端扩展的回文串，即 $S[l..r]$ 是回文串，且 $l = 1$ 或 $r = n$ 或 $S_{l-1} \neq S_{r+1}$。我们可以用 manacher 算法找到 S 中的所有极长回文子串。

建一个 n 个点的图，如果两个点 i, j 在 S 的某个回文串中位于对称的位置，在点 i 和点 j 间连一条表示相等关系的边。对于一个 $l \neq 1, r \neq n$ 的极长回文子串 $S[l..r]$，在 $l-1$ 和 $r+1$ 之间连一条表示不等关系的边。

可以发现，原问题等价于：有多少种方案给这个图的每个点一个 $1 \dots c$ 的颜色，使得每条相等边两端的点颜色相同，不等边两端的点颜色不同。

对不等边作容斥，转化为只有相等边的情况。对只有相等边的情况，假设图中有 k 个连通块，则方案数为 c^k。

可以通过 Subtask 2，获得 20 分。

2.3 算法三

引理 2.1. *对于两个长为 n 的字符串 A 和 B，如果对于任意的 $1 \le i \le n$，$A[1..i]$ 的最长回文子串和 $B[1..i]$ 的最长回文子串长度相等，那么 A 和 B 回文等价。*

Proof. 对字符串的长度作归纳。

$n = 1$ 时，任意两个字符串都回文等价，命题显然成立。

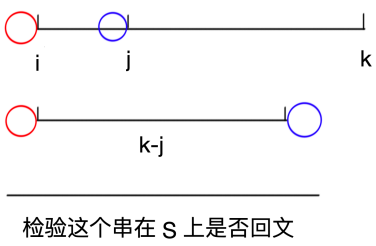
假设 $n = N$ 时，上述命题成立，接下来证明 $n = N + 1$ 时上述命题也成立。设 A 中的最长回文后缀是 $[j..N+1]$。显然，对于 $k < j$，$A[k..N+1]$ 和 $B[k..N+1]$ 都不是回文串。所以只需证明对于 $k > j$，$A[k..N+1]$ 是回文串当且仅当 $B[k..N+1]$ 是回文串即可。对于 $k > j$，$A[k..N+1] = A[j, j+N+1-k], B[k..N+1] = B[j, j+N+1-k]$，因为 $j+N+1-k \le N$，

根据假设，$A[j, j+N+1-k]$ 是回文串当且仅当 $B[j, j+N+1-k]$ 是回文串。因此 $A[k..N+1]$ 是回文串当且仅当 $B[k..N+1]$ 是回文串。所以 A 与 B 回文等价。 □

考虑从前往后决定每一个位置的字符，使得当前已经决定的前缀部分和字符串 S 中对应部分回文等价。决定一个新的字符时，我们只需要保证决定这个字符之后的最长回文后缀与 S 的对应前缀的最长回文后缀相等即可。设我们现在决定的是第 i 个字符，$S[1..i]$ 的最长回文后缀是 $S[j..i]$，若 $j < i$，则第 i 个字符必须与第 j 个字符相等，否则，对于任意一个回文串 $S[k..i-1], k > 1$，都有 $s_{k-1} \neq s_i$。只要满足这个限制，决定第 i 个位置的字符之后得到的串一定与 $S[1..i]$ 回文等价。

总结一下，设你正在决定的字符串为 T，如果 $j < i$，T_i 是确定的，只有 1 种方案；当 $j = i$ 时，设 $P = \{p \mid S[p+1..i-1] = S[p+1..i-1]^R\}$，对于任意的 $j \in P$，$T_i \neq T_j$，下面我们将证明：对于任意的 $x, y \in P$，T_x 与 T_y 是否相等是确定的。（即只与 S 有关，与 T 中已经决定的字符无关）从而决定 T_i 填什么字符的方案数也是确定的。

引理 2.2. *对于三个整数 $1 < i < j \leq k$，如果 $S[i..k]$ 和 $S[j..k]$ 都是回文串，那么要么对于任意与 S 回文等价的字符串 T，T_{i-1} 与 T_{j-1} 相等当且仅当 S_{i-1} 与 S_{j-1} 相等。*



上述引理成立的原因很容易理解，这里只放一张图，证明就不写了。

因此只要从前往后决定每一个位置的字符，计算一下当前位置的字符有多少种填法，乘起来就是答案，这样就得到了一个多项式复杂度的算法。

同时这也说明了，用表示相等关系的边缩点之后可以得到一个弦图，其完美消除序列即把每个点按其对应的缩点前的连通块编号最小的点从小到大排列。如果你没有注意到这一点，也可以解决本题。

2.4 算法四

考虑把算法三中的过程做到 $\mathcal{O}(n)$。

设 p_i 表示点 i 由相等边可以走到的编号最小的点，考虑如何求出 p 数组。

跑一遍 manacher，显然，对 manacher 中的每次比较操作，根据比较结果连上一条相等边或者不等边得到一个图 G，图 G 的一种合法的在每个点上填上一个 $1 \dots c$ 的方案就对应一个与 S 回文等价的字符串。而 manacher 中的比较操作次数是 $\mathcal{O}(n)$ 的，我们可以把这个图建出来然后 dfs。

事实上存在更方便的实现，在做 manacher 时，得到的相等关系 $i < j, T_i = T_j$ 中 j 是每次递增的。因此我们初始时把 p_i 设为 i，每当找到一对相等关系 $i < j, T_i = T_j$ 时，把 p_j 设为 p_i 即可。这样就直接得到了 p 数组。

通过 manacher，我们可以得到 $\mathcal{O}(n)$ 个 S 的全部极长回文串。显然如果 $S[1..i]$ 的最长回文后缀长度为 1，那么 $S[1..i-1]$ 的每个回文后缀都是极长回文串。我们可以开个 vector 记下以每个点为右端点的所有极长回文串。在算 T_i 有多少种填法时只需要检查一下以 $i-1$ 为右端点的所有极长回文串的左端点左边一个点有多少种不同的 p 即可。

这个算法的时间复杂度为 $\mathcal{O}(n)$，可以通过所有测试点，获得 100 分。