

NOIP 2019 模拟赛 Day 1

diamond_duke

题目名称	分组	折纸	集合
可执行文件名	group	origami	set
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
时间限制	1s	1s	1s
内存限制	512MB	512MB	512MB
子任务个数	5	4	5
题目类型	传统型	传统型	传统型

请注意： 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

1 分组

1.1 子任务 3

先假设所有人的重要程度都不一样，那么按照重要程度从小到大排序之后，组长一定在组员的后面。

考虑 DP：设 $f_{i,a,b}$ 表示考虑了排序后的前 i 个人，目前已经有 a 个组配好了，还有 b 个组只有组员没有组长的最小代价。转移时，考虑当前的人是作为组长，加入一个已经有组员的组，还是作为组员新建一个组即可。

然后对于有的人重要程度相同的情况，我们需要想办法继续保证组长在组员的后面。则对于重要程度相同的两个人，我们按照他们的志愿排序：先是只能当组员的，然后是都可以的，然后是只能当组长的。

时间复杂度： $\Theta(nk^2)$ ，可以通过前 3 个子任务。

1.2 满分做法

注意到若 $2k > n$ ，则一定无解。因此我们只要考虑 $2k \leq n$ 的情况。设 $nk = r \leq 10^5$ ，则此时一定有 $k \leq \sqrt{r}$ 。

因此，在特判了 $2k > n$ 的情况后，该做法的时间复杂度实际上为 $\Theta(r\sqrt{r})$ 。

2 折纸

可以发现折纸之后被折到上面的部分实际上是没有用的，因为他和下面对应位置一定是一样的，而影响答案的只有每个位置的颜色和最底层的坐标范围。因此，我们只需要考虑最底层即可，即我们可以把折纸等效为裁纸，每次去掉较小的那一部分。

2.1 子任务 2

最后剩下的底层一定是原来字符串的一个区间 $[l, r]$ 。因为我们将折纸等效为了裁纸，所以向左折和向右折实际上是独立的： $[l, r]$ 可以作为最后剩下的区间，等价于 $[l, n]$ 和 $[1, r]$ 都可以作为最后剩下的区间。

考虑如何对于每个 $i \in [1, n]$ ，求出 $[i, n]$ 以及 $[1, i]$ 能否作为最后剩余的区间。显然这两种情况是对称的，在这里我们只考虑如何求出前者。

考虑使用 Manacher 算法预处理出任意两个位置之间作为中心点时，极长回文子串的长度。则我们考虑当前的位置 i ，我们得要把之前的位置都给折过来。

设上一次折的位置是 j ，则合法等价于 $i - 0.5$ 作为中心点时，回文串长度至少为 $2(i - j)$ ，且 $i - j \leq n - i + 1$ ，以及 $[j, n]$ 可以折出来。可以发现第一个条件蕴含第二个条件，因此我们只要维护最大的，可以折出来的 j 即可。

时间复杂度： $\Theta(m)$ 。

2.2 满分做法

我们可以发现，横着折并不会影响竖着折；反之亦然。这是因为折纸要求对应位置是一样的，那么如果横着折之后可以竖着折，则展开后应当也是对称的，即同样也可以折；反之亦然。

那么我们可以分别计算出横着折以及竖着折的位置，然后相乘就是最后的答案。

在把每一行、每一列进行哈希之后，横着折以及竖着折都是一个一维的问题了，直接套用子任务 2 的做法即可。

时间复杂度： $\Theta(nm)$ （瓶颈在于读入）。

3 集合

3.1 子任务 2 & 3

如果没有异或操作或者没有整体加一的操作，那么我们直接用一个 `std::set` 维护原来的值，然后维护一个懒标记状物即可。

我们以异或操作为例，我们可以维护一个懒标记 y ，初始为 0。当加入或删除 x 时，我们将 $x \oplus y$ 对应地插入 `set` 或从 `set` 中移除。当异或 x 时，我们令 y 变为 $x \oplus y$ 即可。

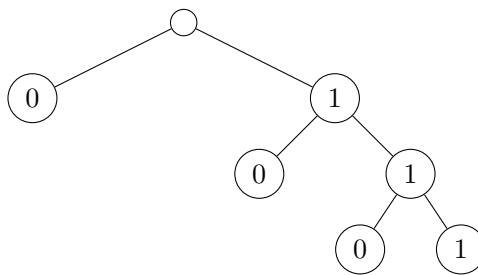
时间复杂度： $\Theta((n + q) \log_2 a_i)$ 。

3.2 满分做法

我们同样维护一个异或操作的懒标记 y ，并对应地处理插入、删除，以及异或操作。接下来的问题在于如何处理整体加一。我们先考虑没有懒标记的情况下如何完成这一操作。

考虑将一个数字 x 变为 $(x+1) \bmod 2^{30}$ 的过程，设 x 在二进制表示下从低到高位依次为 a_1, a_2, \dots, a_{30} ，则我们应当找到一个最小的 i ，使得 $a_1 = a_2 = \dots = a_{i-1} = 1$ ，且 $a_i = 0$ ，然后将 a_1, a_2, \dots, a_i 翻转。如果不存在这样的 i ，则认为 $i = 31$ ，此时需要把全 1 变为全 0。

那么我们考虑将所有数字按照从低到高位顺序插入一棵字典树，在加一时，考虑枚举这样的 i 。容易发现，满足这样条件的 x 对应了字典树的一棵子树，而这些子树的根节点对应了字典树中的一条链上面挂着的一些子树，即图中的所有 0 以及最下面的 1：



翻转操作对应字典树中的交换左右孩子。于是，我们找出这条 1 组成的链后，依次交换链上所有左右孩子即可。

对于有懒标记的情况，唯一的不同点在于这条链不一定全为 1：实际上应当为 $(2^{30} - 1) \oplus y$ 。类似操作即可。

时间复杂度： $\Theta((n + q) \log_2 a_i)$ 。