

省选联考 2024 模拟赛

题目名称	幸运数字	神秘排列	猫猫聚会
题目类型	传统型	传统型	传统型
目录	lucky	perm	cats
可执行文件名	lucky	perm	cats
输入文件名	lucky.in	perm.in	cats.in
输出文件名	lucky.out	perm.out	cats.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	1024 MiB	512 MiB
测试点数目	10	10	10
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	lucky.cpp	perm.cpp	cats.cpp
-----------	-----------	----------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项（请仔细阅读）：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
3. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 程序可使用的栈内存空间限制与题目的内存限制一致。
5. 提交的源程序文件大小不得超过 50 KiB。
6. 只提供 Linux 格式附加样例文件，因系统差异造成的问题请自行处理。
7. 保证各题时限在 std 所用最大时间的两倍以上。
8. 题目较简单，请喧哗的同学不要大声 AK。

幸运数字 (lucky)

【题目描述】

定义所有在十进制表示下仅由 4 和 7 组成的数字是幸运的。例如，474 和 77 是幸运数字，而 456 不是。

Strawberry 和 Icecream 在玩游戏，她们面前有一个由正数组成的序列 $l, l + 1, \dots, r - 1, r$ 。

游戏规则如下：先由 Strawberry 在序列中选择一个长度为 A 的连续区间，再由 Icecream 在 Strawberry 选择的区间中选择一个长度为 B 的区间。

她们这次游戏的得分为 Icecream 选择的区间中幸运数字的个数。

其中 Strawberry 想最大化这个游戏的得分，而 Icecream 想最小化。

在两人都足够聪明的情况下，请输出游戏的结果。

【输入格式】

从文件 *lucky.in* 中读入数据。

输入共一行四个整数 l, r, A, B ，表示 $[l, r]$ 组成的序列，Strawberry 和 Icecream 分别选择的区间长度。

【输出格式】

输出到文件 *lucky.out* 中。

输出一行一个整数，表示游戏的最终得分。

【样例 1 输入】

1 41 80 21 17

【样例 1 输出】

1 1

【样例 2】

见选手目录下的 *lucky/lucky2.in* 与 *lucky/lucky2.ans*。

【样例 3】

见选手目录下的 *lucky/lucky3.in* 与 *lucky/lucky3.ans*。

【样例 4】

见选手目录下的 *lucky/lucky4.in* 与 *lucky/lucky4.ans*。

【子任务】

对于 100% 的测试数据, $1 \leq l \leq r \leq 10^{18}$, $1 \leq B \leq A \leq r - l + 1$ 。

每个测试点的具体限制见下表:

测试点编号	$l, r \leq$
1	500
2	5000
3	10^6
4	
5	10^{10}
6	
7	
8	10^{18}
9	
10	

神秘排列 (perm)

【题目描述】

Strawberry 给你一个序列 p ，初始时它是一个 $1 \sim n$ 的排列。定义一次操作如下：

- 选取序列连续的一段 $[l, r]$ ；
- 求出 $mx = \max(p_l, p_{l+1}, \dots, p_r)$ ；
- 将 p_l, p_{l+1}, \dots, p_r 全部替换为 mx 。

你可以操作这个序列至多 k 次，请求出共能得到多少种不同的序列。答案对 998244353 取模。

【输入格式】

从文件 `perm.in` 中读入数据。

第一行两个整数 n, k 。

第二行 n 个整数，表示排列 p 。

【输出格式】

输出到文件 `perm.out` 中。

一行一个整数，表示答案。

【样例 1 输入】

```
1 3 2
2 3 1 2
```

【样例 1 输出】

```
1 4
```

【样例 2 输入】

```
1 4 2
2 1 2 3 4
```

【样例 2 输出】

```
1 13
```

【样例 3】

见选手目录下的 *perm/perm3.in* 与 *perm/perm3.ans*。

【样例 4】

见选手目录下的 *perm/perm4.in* 与 *perm/perm4.ans*。

【子任务】

对于 100% 的测试数据， $1 \leq n \leq 400, 0 \leq k \leq 400$ ， p 是一个排列。
每个测试点的具体限制见下表：

测试点编号	$n, k \leq$	特殊性质
1	5	无
2		
3	100	
4		
5	400	保证 $p_i = i$
6		
7		无
8		
9		
10		

猫猫聚会 (cats)

【题目描述】

Strawberry 和 Icecream 同时举办了两场聚会。她们有 n 只猫猫，每只猫猫要恰好参加两个聚会中的一个。

一些猫猫有自己的要求，它们只愿意参加其中某一个聚会。

还有几对猫猫之间有矛盾，如果一对有矛盾的猫猫参加了同一场聚会，它们会不开心。但是 Strawberry 和 Icecream 学会了贿赂猫猫，如果她们给第 i 只猫猫 a_i 个金币，这只猫猫就不会来参加任何一场聚会，也不会产生矛盾。

输入数据保证在不考虑猫猫自己的要求的情况下，存在一种划分方案使得没有任何一对有矛盾的猫猫在同一个聚会。

现在你要帮她们花费最少的金币，使得所有猫猫都满足了自己的要求（或通过金币使它没有参加聚会），且没有矛盾出现。

【输入格式】

从文件 `cats.in` 中读入数据。

输入共 $m + 3$ 行。

第一行两个整数 n, m ，表示参加聚会的猫猫数和有矛盾的猫猫对数。

第二行 n 个整数 a_1, a_2, \dots, a_n ，表示让一只猫猫不参加聚会需要的金币数。

第三行 n 个整数 t_1, t_2, \dots, t_n ，表示每只猫猫的要求。 $t_i = 1$ 表示只能参加 Strawberry 的聚会， $t_i = 2$ 表示只能参加 Icecream 的聚会， $t_i = 3$ 表示没有限制。

接下来 m 行，每行两个整数 u_i, v_i ，表示猫猫 u_i 和 v_i 之间有矛盾。

【输出格式】

输出到文件 `cats.out` 中。

输出一行一个整数，表示最少需要花费的金币数。

【样例 1 输入】

```
1 3 2
2 2 1 2
3 1 3 2
4 1 2
5 2 3
```

【样例 1 输出】

11

【样例 2 输入】

17 6
28 10 7 3 3 10 1
31 2 3 1 3 1 1
43 5
54 3
61 2
74 1
87 1
93 2

【样例 2 输出】

14

【样例 3】

见选手目录下的 *cats/cats3.in* 与 *cats/cats3.ans*。

【样例 4】

见选手目录下的 *cats/cats4.in* 与 *cats/cats4.ans*。

【子任务】

对于 100% 的测试数据， $1 \leq n \leq 10^4, 0 \leq m \leq 10^5, 1 \leq a_i \leq 10^5, t_i \in [1, 3], 1 \leq u_i, v_i \leq n$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	特殊性质
1	15	20	无
2	200	500	
3			
4	10^4	10^5	保证 $t_i \in [1, 2]$
5			保证 $a_i = 1$
6			
7			
8			无
9			
10			