NOIP 2020模拟赛

考试时间:4.5小时

题目名称	铺设积木大赛	豪华湖心岛	奇怪的众数	皇后平板
源文件名	match.cpp	island.cpp	mode.cpp	hhpb.cpp
时间限制	3s	1s	3s	1s
内存限制	512MB	512MB	512MB	512MB
编译选项	-std=c++11	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11

时限可根据评测机速度调整为std1.5倍以上。

铺设积木大赛(match)

题目描述

M城将举行一场铺设积木大赛。在比赛中，选手们需要运用自己的才智，搭建出一个由积木组成的道路。

LJF007 参加了这次大赛。她打算用积木铺一条起伏的道路，以显示命途艰辛。经过初步规划，这条积木之路将在一块平地上搭建，它有 n 格长，在第 i 格需要放置 h_i 块积木。初始平地上没有任何积木。

积木的搭建遵循以下规则：在第 i 次可以选取一个合法区间 $[l,r]$，以消耗体力$i\times(r-l+1)$的代价，在区间上的每格放置 1 块积木。一个区间合法，当且仅当区间中每格的高度相同，且都小于目标高度。

LJF007 不想消耗太多体力，您需要告诉她最小的体力总消耗值。

输入格式

本题有多组数据，且有特殊输入方式。

第一行给出 2 个64位有符号整数 S,P 用于特殊读入。

第二行为 1 个正整数数 T，表示数据组数。

对于每组数据包含一个正整数 n。接下来你要调用 n 次rnd函数：

```
#define LL long long
LL S,P;
int rnd() {
    S^=S<<13,S^=S>>17,S^=S<<5;
    return (S%P+P)%P;
}

```

每次调用得到的值表示该位置应铺设的积木高度。

输出格式

T行，表示每组数据的答案。

输入输出样例

sample1.in

```
666 233
2
2
5
```

sample1.out

```
18351
64772
```

sample2.in

```
326453276652 1000000007
2
1000000
5000
```

sample2.out

```
6257362517594013314
10745364984604095562
```

样例解释

样例1第一组数据的高度分别为$5,191$，不难发现答案是$\frac{5\times 6}{2}\times 2+\frac{(6+6+185)\times 186}{2}=18351$

样例1第二组数据的高度分别为$193,214,128,155,27$，我有一个绝妙的解释，可惜这里写不下。

数据范围

对于前 10% 的数据满足 $P=2$

对于前 30% 的数据满足 $n\leq 1000$。

对于前 60% 的数据满足 $n\leq 50000$。

对于前 80% 的数据满足 $n\leq 10^5$。

对于 100% 的数据满足 $T\leq 30,n\leq 10^6,1\leq s,c<2^{64},0\leq h_i<P\leq 10^9+7$，保证 P 是质数。

答案较大，请使用ULL自然溢出。

豪华湖心岛(island)

题目描述

已知在大陆上有n个城市，这些城市由m条双向道路连接，每条道路长度为1，保证无重复道路，保证这n个城市互相连通。有k个国家控制着这n座城市，第i个国家的首都为K_i。初始(t=0)时，在第i个城市，有x_i名士兵驻守该城，并且在每个单位时刻，该城都会增加y_i名士兵（于战争和特殊手段之后发生）。

众所周知，各个国家之间会发生战争。一次战争的形式如下:在t_i时刻，一个国家将会从a_i出发，攻打城b_i（若a_i和b_i属于同一国家，则视为战争无效，也不会集结兵力。保证有道路连接a_i和b_i），进攻方会集结距离a_i为1的所有可达城市的士兵于a_i（记总数为sum_a），防守方会集结距离b_i为1的所有可达城市的士兵于b_i（记总数为sum_b）。（注意，不会集结盟国士兵，只能集结本国士兵）

$if\ sum_a \leq sum_b$ 进攻失败，a_i剩余士兵为0，b_i剩余士兵为$sum_b - sum_a$。

$if sum_a > sum_b$进攻成功，b_i被进攻方占领，a_i剩余士兵为0，b_i剩余士兵为$sum_a - sum_b$。

特殊的，如果一个国家的首都被A国攻占，那么该国家的所有城市全部归属于A国，之后所有关于该国的操作视为无效。

特殊的，如果一个城市与该国的首都不能在不经过其它国家城市的情况下互相到达，那么该城会得到$debuff$孤城：在每个单位时间末，该城不会再增加士兵。

事实上，国与国之间还会有几种特殊手段，在t_i时刻，国家c_i会使用第pi种策略：

1.联盟：给出d_i，保证$c_i \neq d_i$。该国将会与d_i国家结盟，使得在接下来的时间中，两国之间所有的手段和战争无效，两国可能重复结盟。（需要注意的是，联盟不具有传递性，如果国家1与国家2、国家3联盟，那么国家2和3之间依然有可能出现手段和战争）

2.空袭：给出$T,\ a_i$。该国会对城市a_i采用空袭，(如果该城属于结盟国家或c_i，则视为无效)，该国的驻守士兵减半(向下取整)，同时在T时刻之前无法再增加士兵。保证$T > t_i$。

3.摧毁道路：给出$T,\ a_i,\ b_i$。该国会在T时刻将城市a_i和b_i间的道路摧毁（优先级最高，如果在T时刻a_i和b_i中有至少一座城属于结盟国家或c_i，则视为无效，注意不考虑当前时间t_i），保证n个城市仍然互相连通。保证$T > t_i$。

4.强行军令：该国在下一次遭遇战争的时候，可以集结距离小于等于2的城市的士兵于战争城市，之后强行军令失效。强行军令可以累加，但是范围不会扩大。

5.迁都：给出a_i（如果a_i不属于c_i，则视为无效）。该国将首都迁到a_i，不保证a_i与原来的首都可以不经过其他国家城市互相到达。

6.征兵令：给出x_i。该国首都增加x_i名士兵。

现在想要你求最终每个国家的城市数目。

输入格式

第一行三个整数n,m,k,q分别表示城市总数，道路总数，国家总数，战争和特殊手段总数。

第二行n个数，第i个数k_i表示第i个城市属于国家k_i。

第三行n个数，第i个数x_i表示第i个城市初始士兵数。

第四行n个数，第i个数y_i表示每个单位时间末会增加的士兵数。

第五行k个数，第i个数K_i表示第i个国家的首都为K_i。

接下来m行每行两个数u,v，表示城u和城v之间有道路连接。

接下来q行，每行代表一次战争或特殊手段，输入格式如下：

读入一个数op，如果op为0表示为战争，否则为特殊手段。

对于战争读入三个数t_i，a_i，b_i，意义如题所示。

对于特殊手段读入三个数t_i，c_i，p_i，意义如题所示。

$p_i = 1$ 读入一个数d_i。

$p_i = 2$ 读入两个数T，a_i。

$p_i = 3$ 读入三个数T，a_i，b_i。

$p_i = 5$ 读入一个数a_i。

$p_i = 6$ 读入一个数x_i。

输出格式

输出i个数，分别表示每个国家拥有的城市数。

输入输出样例

sample1.in

```
7 9 2 3

1 1 1 1 2 2 2

9 1 4 7 2 7 3

2 3 1 1 1 2 1

2 7

1 2

1 4

2 3

3 4

4 5

4 6

5 7

6 7

5 6

0 3 5 4

1 2 2 6 10

0 1 5 4
```

sample1.out

```
3 4
```

样例解释

t=0时，各城市士兵数分别为(9,1,4,7,2,7,3)。

t=1时，战争前士兵数分别为(0,1,0,20,12,0,0)。

战争后士兵数分别为(0,1,0,8,0,0,0)。

招兵后士兵数分别为(2,4,1,9,1,2,1)。

t=2时，手段后士兵数分别为(2,4,1,9,1,2,11)。

招兵后士兵数分别为(4,7,2,10,2,4,12)。

t=3时，战争前士兵数分别为(0,7,0,16,18,0,0)。

战争后士兵数分别为(0,7,0,2,0,0,0)，并且城4属于国家2。

城1,2,3,属于国家1，城4,5,6,7属于国家2。

sample2.in

```
9 14 3 21
2 1 1 1 2 3 3 3 2
1 3 4 2 5 6 1 3 7
1 4 2 5 7 3 5 7 2
2 5 6
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
2 3
3 4
4 5
6 7
7 8
8 9
0 2 3 1
0 3 6 1
1 4 2 6 20
0 5 5 1
1 6 2 6 50
1 7 2 6 70
0 8 5 1
1 9 1 4
1 10 2 1 3
0 11 7 1
0 12 4 5
1 13 3 2 15 4
1 14 3 3 15 3 4
0 15 7 1
1 16 3 4
0 17 1 9
1 18 1 5 5
1 19 1 6 100
0 20 1 5
0 21 5 1
0 22 1 6
```

sample2.out

```
4 0 5
```

下发文件中给出了 10 个样例。

其中前 3 个为普适性样例，后 7 个是满足特殊性质的样例，请根据文件名区分。

数据规模

保证t_i互不相同，保证连边随机。

对于前10%的数据$k=1$。

对于前40%的数据$op=0$。

对于另外10%的数据$p_i=1$。

对于另外5%的数据$p_i=2$。

对于另外5%的数据$p_i=3$。

对于另外10%的数据$p_i=4$。

对于另外5%的数据$p_i=5$。

对于另外5%的数据$p_i=6$。

对于100%的数据$n\leq 500$，$m\leq 1000$，$u\leq n$，$v\leq n$，$k\leq n$，$q\leq 1000$，$x_i\leq 1000$，$y_i\leq 10$，$t_i\leq 10^9$。

奇怪的众数(mode)

题目描述

给出 2 个长为 $n-1$ 的序列（**保证n为质数**） $a_1\dots a_{n-1}$ 和 $b_1\dots b_{n-1}$，请你任意排列这两个序列，使得序列 $a_1\times b_1 \pmod n \dots a_{n-1}\times b_{n-1} \pmod n$ 的众数的出现次数最大。为了方便，你只需要输出这个最大的出现次数。

输入格式

第一行 1 个正整数 n。

第二行 n 个正整数 $a_1\dots a_{n-1}$。

第三行 n 个正整数 $b_1\dots b_{n-1}$。

输出格式

1 个整数，表示任意排列后最大的众数出现次数。

样例

sample1.in

```
3
1 2
2 1
```

sample1.out

```
2
```

sample2.in&sample2.out

见下发文件。

数据范围

subtask1(20pts)，满足 $n \leq 5000$。

subtask2(20pts)，满足 $n \leq 40000$。

subtask3(10pts)，满足序列 a 中的数互不相等且序列 b 中的数互不相等。

subtask4(20pts)，满足任何一个数在序列 a, b 中出现次数都不超过5（分别计算，即可以各出现5次）。

subtask5(30pts)，无特殊限制。

对于 100% 的数据，满足 $2 \leq n \leq 100003$ 且 n 是质数，$1 \leq a_i, b_i < n$。

皇后平板(hhpb)

题目描述

NOIP也变成了4题，真是一件悲伤的事情。

定义一个 $n \times n$ 的矩阵称为“n阶皇后矩阵”，当且仅当矩阵中的元素都取自集合 $S = \{1, 2 \dots 2n-1\}$，且对于每个 $i = 1, 2 \dots n$，它的第 i 行和第 i 列中所有元素合起来恰好是 S 中的所有元素。

而我们定义一个边长大于等于 m 的皇后矩阵为“皇后平板”，这个 m 随皇后的心情变化，因此请你对于给定的 m 给出一个皇后平板。当然如果你不想回答这个问题，你也可以退而求其次，回答"是否存在m阶皇后矩阵"。

由于一些原因，你输出的矩阵边长不能超过 2048。

输入格式

一行，一个正整数 m ，表示边长大于等于 m 的矩阵才能称之为皇后矩阵。

输出格式

你可以输出 2 种类型的答案，因此第一行你需要输出一个整数 $op(op = 1$ 或 $2)$，接下来：

- 若 $op = 1$，说明你愿意回答"是否存在m阶皇后矩阵"，如果存在，请在第二行输出YES，如果不存在，请在第二行输出NO。如果回答正确，你可以获得该测试点30%的分数。
- 若 $op = 2$，说明你愿意给出一个边长大于等于 m 的皇后矩阵，请在第二行输出一个整数 n 表示矩阵的边长，接下来 n 行，每行 n 个整数表示第 i 行第 j 个位置的数的权值。

样例

sample1.in

```
1
```

sample1.out

```
2
1
1
```

注：样例1也可以输出一个边长为2（或更大）的皇后矩阵

sample2.in

```
1997
```

sample2.out

```
1
NO
```

注：样例2的输出只能获得30%的分数

数据范围

对于前20%的测试点，$m \leq 4$。
对于前40%的测试点，$m \leq 8$。
对于前60%的测试点，$m \leq 20$。
对于前100%的测试点，$m \leq 2000$。

对于你的输出：

如果 $op = 1$ 且输出正确，你将获得该测试点30%的分数。

如果 $op = 2$ 且输出正确且矩阵边长不大于 2048，你将获得该测试点 100% 的分数。