

NOIp信心赛

2016 年 11 月 1 日

题目名称	Introduc	Development	Closing
目录	introduc	development	closing
可执行文件名	introduc	development	closing
输入文件名	introduc.in	development.in	closing.in
输出文件名	introduc.out	development.out	closing.out
每个测试点时限	1秒	1秒	1秒
内存限制	128MB	128MB	128MB
测试点数量	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型

提交源程序须加后缀

对于 C++ 语言	introduc.cpp	development.cpp	closing.cpp
对于 C 语言	introduc.c	development.c	closing.c
对于 Pascal 语言	introduc.pas	development.pas	closing.pas

本场比赛在`LINUX`下评测,但数据由`Windowss`环境构造,请注意读入输出问题.

Introduct

【问题描述】

小C非常喜欢吊打对手,所以他发明了一款游戏,并且来找你一起玩.

小C的游戏在一棵树(OI意义)上进行,编号为 $0 \sim N-1$,你作为红方,小C作为蓝方.每个人在树上的拥有若干个棋子,每个棋子都在不同的位置上.每轮游戏中,红方先动.每次行动中每个人都可以把所有棋子向任意一个方向走一步,也可以不走.红方的目的是尽量躲避,蓝方的目的是尽量快的抓到至少一个棋子,只要有一个棋子被抓,游戏就结束了.可以发现蓝方一定会胜利.现在请你求出在两方都采取最优策略下,游戏会进行几轮.

【输入格式】

从文件 `introduct.in` 中读入数据。

第一行三个整数 N, M, K ,描述树的大小,以及红/蓝方的棋子数目.

第二行 $N-1$ 个整数 f_{a_i} ,描述 $1 \sim N-1$ 的父亲.

第三行 M 个整数 x ,描述红方棋子的位置.

第四行 K 个整数 y ,描述蓝方棋子的位置.

【输出格式】

输出到文件 `introduct.out` 中。

一行输出答案.

【样例输入1】

```
7 1 2
0 0 0 3 4 2
1
6 5
```

【样例输出1】

3

【数据规模】

对于50%的数据: $N, M, K \leq 5$.

对于100%的数据: $N, M, K \leq 50, x, y$ 互不相同, $0 \leq fa_i \leq i - 1$.

Development

【问题描述】

定义对于一颗 M 个节点的树 T , $S(T) = \sum_{i=1}^M \sum_{j=i+1}^M dis(i, j)$, 其中 $dis(i, j)$ 为 i 到 j 的最短距离.

现在给出 N, P, R , 要求求出一个最小的 $S(T)$, 满足 $|T| = N$, 且 $S(T) \bmod P = R$. 如果无解输出 -1 .

【输入格式】

从文件 development.in 中读入数据。

一行 3 个整数 N, P, R .

【输出格式】

输出到文件 development.out 中。

一行描述答案。

【样例输入一】

6 2 0

【样例输出一】

28

【数据规模】

对于 30% 的数据 $N \leq 8$.

对于 50% 的数据 $N \leq 20$

对于 100% 的数据: $2 \leq N \leq 50, 1 \leq P \leq 100, 0 \leq R < P$.

Closing

【问题描述】

小 C 有一张完全二分图(每条边都存在),左边有 N 个点,右边有 M 个点.

小 C 想在这张图上玩一个游戏.他从 dj 那里拿来了 N 个红色棋子和 M 个蓝色棋子,分别标号为 $R_0 - - R_{N-1}$ 和 $B_0 - - B_{M-1}$.现在他把 N 个红色棋子任意放到左边 N 个节点上, M 个蓝色棋子任意放到右边 M 个节点上.现在他希望能通过一些操作使得操作后左边 N 个节点上依次为 $R_0 - - R_{N-1}$,右边 M 个节点上依次为 $B_0 - - B_{M-1}$.一次操作定义为交换一条边相邻的两个棋子.

我要在1000步之内完成交换,小 C 想.然后他发现这实在是太简单了.

每条边只能用一次,小 C 想.然后他发现这似乎并不太可做.

最多只能有一条边用两次,小 C 想.然后他发现这个游戏非常有趣.

现在小 C 告诉你初始每个节点上的棋子编号,请你构造出一组解。

【输入格式】

从文件 closing.in 中读入数据。

第一行两个正整数 N, M .

第二行 N 个整数 A_i ,描述对于 $0 \leq i < N$,左侧第 i 个位置上的棋子是 R_{A_i} .

第三行 M 个整数 C_i ,描述对于 $0 \leq i < M$,右侧第 i 个位置上的棋子是 B_{C_i} .

【输出格式】

输出到文件 closing.out 中。

第一行一个整数 ANS 描述你方案的长度.

接下来 ANS 行,每行两个正整数 x_i, y_i ,表示交换左边第 x_i 个位置和右边第 y_i 个位置的棋子.

如果无解,输出且仅输出一行-1.

【样例输入1】

2 2

1 0

1 0

【样例输出1】

4

0 1

1 0

0 0

1 1

【数据规模】

对于10%的数据, $N + M \leq 5$.

对于另20%的数据, $A_i = (i + 1) \bmod N, B_i = (i + 1) \bmod M$.

对于另20%的数据, $A_i = i$.

对于100%的数据: $N, M \leq 100, N, M \geq 2$.