

NOI2019 湖南省组队选拔

第二试试题

一. 题目概况

题目名称	校园旅行	白兔之舞	序列
目录	tour	dance	sequence
可执行文件名	tour	dance	sequence
输入文件名	tour.in	dance.in	sequence.in
输出文件名	tour.out	dance.out	sequence.out
每个测试点时限	2 秒	2 秒	2 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
结果比较方式	字符串比较, 多行单字符串比较	整数比较, 多行单个数字比较	整数比较, 多行单个数字比较
题目类型	传统	传统	传统
内存上限	512	512M	512M

二. 提交源程序需加后缀

对于 C++ 语言	tour.cpp	dance.cpp	sequence.cpp
对于 C 语言	tour.c	dance.c	sequence.c
对于 Pascal 语言	tour.pas	dance.pas	sequence.pas

三. 编译命令

对于 C++ 语言	g++ -o tour tour.cpp -lm -O2	g++ -o dance dance.cpp -lm	g++ -o sequence sequence.cpp -lm -O2
对于 C 语言	gcc -o tour tour.c -lm -O2	gcc -o dance dance.c -lm	gcc -o sequence sequence.c -lm -O2
对于 Pascal 语言	fpc tour.pas -O2	fpc dance.pas	fpc sequence.pas -O2

注意事项:

- (1) 选手必须在自己的工作目录下操作, 严禁在其他目录下工作。目录结构请遵从 NOI 规范, 即需要在工作目录下再为每个题目建相应子目录, 子目录名为对应题目的英文名。
- (2) 选手最后提交的源程序 (.pas 或 .c 或 .cpp) 必须在自己的工作目录里对应子目录下, 对于缺少文件者, 不予测试, 该题计零分。
- (3) 子目录名、源程序文件名和输入输出文件名必须使用英文小写。
- (4) 特别提醒: 评测在 NOI Linux 下进行。

第 1 题：校园旅行(tour)，运行时限 2s，内存上限 512M，100 分。

【问题描述】

某学校的每个建筑都有一个独特的编号。一天你在校园里无聊，决定在校园内随意地漫步。

你已经在校园里呆过一段时间，对校园内每个建筑的编号非常熟悉，于是你情不自禁的把周围每个建筑的编号都记了下来——但其实你没有真的记下来，而是把每个建筑的编号除以 2 取余数得到 0 或 1，作为该建筑的标记，多个建筑物的标记连在一起形成一个 01 串。

你对这个串很感兴趣，尤其是对于这个串是回文串的情况，于是你决定研究这个问题。

学校可以看成一张图，建筑是图中的顶点，而某些顶点之间存在无向边。对于每个顶点我们有一个标记（0 或者 1）。每次你会选择图中两个顶点，你想知道这两个顶点之间是否存在一条路径使得路上经过的点的标记形成一个回文串。

一个回文串是一个字符串使得它逆序之后形成的字符串和它自己相同，比如“010”，“1001”都是回文串，而“01”，“110”不是。注意长度为 1 的串总是回文串，因此如果询问的两个顶点相同，这样的路径总是存在。此外注意，经过的路径不一定为简单路径，也就是说每条边每个顶点都可以经过任意多次。

【程序文件名】

源程序文件名为 tour.cpp/c/pas。

【输入格式】

输入文件名为 tour.in。

第一行三个整数 n, m, q ，表示图中的顶点数和边数，以及询问数。

第二行为一个长度为 n 的 01 串，其中第 i 个字符表示第 i 个顶点（即顶点 i ）的标记，点从 1 开始编号。

接下来 m 行，每一行是两个整数 u_i, v_i ，表示顶点 u_i 和顶点 v_i 之间有一条无向边，不存在自环或者重边。

接下来 q 行，每一行存在两个整数 x_i, y_i ，表示询问顶点 x_i 和顶点 y_i 的点之间是否有一条满足条件的路径。

【输出格式】

输出文件名为 tour.out。

输出 q 行，每行一个字符串“YES”，或者“NO”（引号不输出）。输出“YES”表示满足条件的路径存在，输出“NO”表示不存在。

【输入输出样例 1】

tour.in	tour.out
5 4 2	NO
00010	YES
4 5	
1 3	
4 2	
2 5	
3 5	
1 3	

【样例解释 1】

对于第一个询问，3 号点和 2 号点不连通，因此答案为“NO”。

对于第二个询问，一条合法的路径是 $1 \rightarrow 3$ ，路径上的标号形成的字符串为“00”。注意合法路径不唯一。

【输入输出样例 2】

tour.in	tour.out
10 11 10	NO
0011011111	YES
4 6	YES
10 6	NO
5 9	YES
4 7	YES
10 7	YES
5 8	YES
1 9	YES
5 7	NO
1 10	
5 1	
5 6	
10 3	
7 4	
8 10	
9 4	
8 9	
6 6	
2 2	
9 9	
10 9	
3 4	

【数据范围】

对于 30%的数据， $1 \leq m \leq 10000$ ；

对于 70%的数据， $1 \leq m \leq 50000$ ；

对于 100%的数据， $1 \leq n \leq 3000, 1 \leq m \leq 500000, 1 \leq q \leq 100000$ 。

【编译命令】

对于 c++ 语言： `g++ -o tour tour.cpp -lm -O2`

对于 c 语言： `gcc -o tour tour.c -lm -O2`

对于 pascal 语言： `fpc tour.pas -O2`

第 2 题：白兔之舞(dance)，运行时限 2s，内存上限 512M，100 分。

【问题描述】

有一张顶点数为 $(L+1)*n$ 的有向图。这张图的每个顶点由一个二元组 (u,v) 表示($0 \leq u \leq L, 1 \leq v \leq n$)。这张图不是简单图，对于任意两个顶点 $(u1,v1)(u2,v2)$ ，如果 $u1 < u2$ ，则从 $(u1,v1)$ 到 $(u2,v2)$ 一共有 $w[v1][v2]$ 条不同的边，如果 $u1 \geq u2$ 则没有边。

白兔将在这张图上上演一支舞曲。白兔初始时位于该有向图的顶点 $(0,x)$ 。

白兔将会跳若干步。每一步，白兔会从当前顶点沿任意一条出边跳到下一个顶点。白兔可以在任意时候停止跳舞（也可以没有跳就直接结束）。当到达第一维为 L 的顶点就不得不停止，因为该顶点没有出边。

假设白兔停止时，跳了 m 步，白兔会把这只舞曲给记录下来成为一个序列。序列的第 i 个元素为它第 i 步经过的边。

问题来了：给定正整数 k 和 y ($1 \leq y \leq n$)，对于每个 t ($0 \leq t < k$)，求有多少种舞曲（假设其长度为 m ）满足 $m \bmod k = t$ ，且白兔最后停在了坐标第二维为 y 的顶点？

两支舞曲不同定义为它们的长度 (m) 不同或者存在某一步它们所走的边不同。

输出的结果对 p 取模。

【程序文件名】

源程序文件名为 `dance.cpp/c/pas`。

【输入格式】

输入文件名为 `dance.in`。

第一行五个用空格隔开的整数 n,k,L,x,y,p 。

接下来 n 行，每行有 n 个用空格隔开的整数，第 i 行的第 j 个数表示 $w[i][j]$

【输出格式】

输出文件名为 `dance.out`。

依次输出 k 行，每行一个数表示答案对 p 取模的结果。

【输入输出样例 1】

dance.in	dance.out
2 2 3 1 1 998244353	16
2 1	18
1 0	

【样例解释 1】

$t=0$:

1. 路径长度为0，方案数为1。

2. 路径长度为2，一共有六类路径

$(0,1) \rightarrow (1,1) \rightarrow (2,1)$ 该路径有 $w[1][1]*w[1][1]=4$ 条

$(0,1) \rightarrow (1,1) \rightarrow (3,1)$ 该路径有 $w[1][1]*w[1][1]=4$ 条

$(0,1) \rightarrow (2,1) \rightarrow (3,1)$ 该路径有 $w[1][1]*w[1][1]=4$ 条

$(0,1) \rightarrow (1,2) \rightarrow (2,1)$ 该路径有 $w[1][2]*w[2][1]=1$ 条

$(0,1) \rightarrow (1,2) \rightarrow (3,1)$ 该路径有 $w[1][2]*w[2][1]=1$ 条

$(0,1) \rightarrow (2,2) \rightarrow (3,1)$ 该路径有 $w[1][2]*w[2][1]=1$ 条

答案就为 $1+4+4+4+1+1+1=16$

t=1:

1.路径长度为1，一共有3类路径

(0,1)->(1,1) 该路径有 $w[1][1]=2$ 条

(0,1)->(2,1) 该路径有 $w[1][1]=2$ 条

(0,1)->(3,1) 该路径有 $w[1][1]=2$ 条

2.路径长度为3，一共有3类路径

(0,1)->(1,1)->(2,1)->(3,1) 该路径有 $w[1][1]*w[1][1]*w[1][1]=8$ 条

(0,1)->(1,1)->(2,2)->(3,1) 该路径有 $w[1][1]*w[1][2]*w[2][1]=2$ 条

(0,1)->(1,2)->(2,1)->(3,1) 该路径有 $w[1][2]*w[2][1]*w[1][1]=2$ 条

答案就为 $2+2+2+8+2+2=18$

【输入输出样例2】

dance.in	dance.out
3 4 100 1 3 998244353	506551216
1 1 1	528858062
1 1 1	469849094
1 1 1	818871911

【数据范围】

测试点1,2: $L \leq 100000$

测试点3: $n=1, w[1][1]=1, k$ 的最大质因子为2

测试点4: $n=1, k$ 的最大质因子为2

测试点5: $n=1, w[1][1]=1$

测试点6: $n=1$

测试点7,8: k 的最大质因子为2

对于全部数据:

p 为一个质数, $10^8 < p < 2^{30}$

$1 \leq n \leq 3$

$1 \leq x \leq n$

$1 \leq y \leq n$

$0 \leq w[i][j] < p$

$1 \leq k \leq 65536$, k 为 $p-1$ 的约数

$1 \leq L \leq 10^8$

【编译命令】

对于c++语言: `g++ -o dance dance.cpp -lm`

对于c语言: `gcc -o dance dance.c -lm`

对于pascal语言: `fpc dance.pas`

第 3 题：序列(sequence)，运行时限 2s，内存上限 512M，100 分。

【问题描述】

给定一个长度为 n 的序列 A_1, \dots, A_n ，以及 m 个操作，每个操作将一个 A_i 修改为 k 。第一次修改之前及每次修改之后，都要求你找到一个同样长度为 n 的单调不降序列 B_1, \dots, B_n ，使得 $\sum_{i=1}^n (A_i - B_i)^2$ 最小，并输出该最小值。需要注意的是每次操作的影响都是独立的，也即每次操作只会对当前询问造成影响。为了避免精度问题，我们保证这个最小值是个分数，也即能表示为两个非负整数相除的形式： x/y 。那么你将要输出 $(x * y^{P-2}) \bmod P$ 的值，表示模意义下 x/y 的值。其中 $P = 998244353$ 是一个大质数。

【程序文件名】

源程序文件名为`sequence.cpp/c/pas`。

【输入格式】

输入文件名为`sequence.in`。

第一行两个非负整数 n, m ，代表序列长度和操作数。

第二行有 n 个由空格隔开的正整数，代表序列 A_1, \dots, A_n 。

接下来 m 行每行两个正整数 i, k ，代表将 A_i 修改为 k 。

【输出格式】

输出文件名为`sequence.out`。

输出 $m + 1$ 行每行一个整数，第 i 行输出第 $i - 1$ 次修改后的答案。特别的，第1行应为初始局面的答案。

【输入输出样例】

sequence.in	sequence.out
5 3	28
9 2 4 6 4	2
1 1	4
1 4	26
5 6	

【样例解释】

第一个询问的最优 B 序列为： $\{5\ 5\ 5\ 5\ 5\}$ 。

第二个询问的最优 B 序列为： $\{1\ 2\ 4\ 5\ 5\}$ 。

第三个询问的最优 B 序列为： $\{3\ 3\ 4\ 5\ 5\}$ 。

第四个询问的最优 B 序列为： $\{5\ 5\ 5\ 6\ 6\}$ 。

样例是存在最优方案使 B_i 皆为整数的特殊情况。

【数据范围】

对于前 10%的数据，保证 $n, m \leq 10$ ， $k, A_i \leq 1000$ ，且存在一种最优方案，使得 B_i 皆为整数。

对于前 30%的数据，保证 $n, m \leq 100$ 。

对于另外 20%的数据，保证 $m = 0$ 。

对于另外 20%的数据，保证 $n, m \leq 3 * 10^4$ 。

对于所有数据，保证 $3 \leq n \leq 10^5$ ， $0 \leq m \leq 10^5$ ， $1 \leq k, A_i \leq 10^9$ 。

【编译命令】

对于 c++语言： `g++ -o sequence sequence.cpp -lm -O2`

对于 c 语言： `gcc -o sequence sequence.c -lm -O2`

对于 pascal 语言： `fpc sequence.pas -O2`