2022 CCF 非专业级软件能力认证

CSP-J/S 2022 第二轮认证

提高级

时间：2022 年 10 月 29 日 14:30 ～ 18:30

题目名称	假期计划	策略游戏	星战	数据传输
题目类型	传统型	传统型	传统型	传统型
目录	holiday	game	galaxy	transmit
可执行文件名	holiday	game	galaxy	transmit
输入文件名	holiday.in	game.in	galaxy.in	transmit.in
输出文件名	holiday.out	game.out	galaxy.out	transmit.out
每个测试点时限	2.0 秒	1.0 秒	2.0 秒	3.0 秒
内存限制	512 MiB	512 MiB	512 MiB	1024 MiB
测试点数目	20	20	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	holiday.cpp	game.cpp	galaxy.cpp	transmit.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/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. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 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）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
9. 只提供 Linux 格式附加样例文件。
10. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

假期计划（holiday）

【题目描述】

小熊的地图上有 n 个点，其中编号为 1 的是它的家、编号为 $2,3,\ldots,n$ 的都是景点。部分点对之间有双向直达的公交线路。如果点 x 与 z_1、z_1 与 z_2、……、z_{k-1} 与 z_k、z_k 与 y 之间均有直达的线路，那么我们称 x 与 y 之间的行程可转车 k 次通达；特别地，如果点 x 与 y 之间有直达的线路，则称可转车 0 次通达。

很快就要放假了，小熊计划从家出发去 4 个不同的景点游玩，完成 5 段行程后回家：家 → 景点 A → 景点 B → 景点 C → 景点 D → 家且每段行程最多转车 k 次。转车时经过的点没有任何限制，既可以是家、也可以是景点，还可以重复经过相同的点。例如，在景点 A → 景点 B 的这段行程中，转车时经过的点可以是家、也可以是景点 C，还可以是景点 D → 家这段行程转车时经过的点。

假设每个景点都有一个分数，请帮小熊规划一个行程，使得小熊访问的四个不同景点的分数之和最大

【输入格式】

从文件 ***holiday.in*** 中读入数据。

第一行包含 3 个正整数 n,m,k，分别表示地图上点的个数、双向直达的点对数量、每段行程最多的转车次数。

第二行包含 $n-1$ 个正整数，分别表示编号为 $2,3,\ldots,n$ 的景点的分数。

接下来 m 行，每行包含两个正整数 x,y，表示点 x 和 y 之间有道路直接相连，保证 $1 \le x,y \le n$，且没有重边，自环。

【输出格式】

输出到文件 ***holiday.out*** 中。

输出一个正整数，表示小熊经过的 4 个不同景点的分数之和的最大值。

【样例 1 输入】

```
8 8 1
9 7 1 8 2 3 6
1 2
2 3
3 4
4 5
5 6
```

```
6 7
7 8
8 1
```

【样例 1 输出】

```
27
```

【样例 1 解释】

当计划的行程为 $1 \to 2 \to 3 \to 5 \to 7 \to 1$ 时，4 个景点的分数之和为 $9+7+8+3 = 27$，可以证明其为最大值。

行程 $1 \to 3 \to 5 \to 7 \to 8 \to 1$ 的景点分数之和为 24、行程 $1 \to 3 \to 2 \to 8 \to 7 \to 1$ 的景点分数之和为 25。它们都符合要求，但分数之和不是最大的。

行程 $1 \to 2 \to 3 \to 5 \to 8 \to 1$ 的景点分数之和为 30，但其中 $5 \to 8$ 至少需要转车 2 次，因此不符合最多转车 $k = 1$ 次的要求。

行程 $1 \to 2 \to 3 \to 2 \to 3 \to 1$ 的景点分数之和为 32，但游玩的并非 4 个不同的景点，因此也不符合要求。

【样例 2 输入】

```
7 9 0
1 1 1 2 3 4
1 2
2 3
3 4
1 5
1 6
1 7
5 4
6 4
7 4
```

【样例 2 输出】

```
7
```

【样例 3】

见选手目录下的 ***holiday/holiday3.in*** 与 ***holiday/holiday3.ans***。

【数据范围】

对于所有数据，保证 $5 \leq n \leq 2500, 1 \leq m \leq 10000, 0 \leq k \leq 100$, 所有景点的分数 $1 \leq s_i \leq 10^{18}$。保证至少存在一组符合要求的行程。

测试点编号	$n \leq$	$m \leq$	$k \leq$
$1 \sim 3$	10	20	0
$4 \sim 5$			5
$6 \sim 8$	20	50	100
$9 \sim 11$	300	1000	0
$12 \sim 14$			100
$15 \sim 17$	2500	10000	0
$18 \sim 20$			100

策略游戏（game）

【题目描述】

小 L 和小 Q 在玩一个策略游戏。

有一个长度为 n 的数组 A 和一个长度为 m 的数组 B ，在此基础上定义一个大小为 $n \times m$ 的矩阵 C ，满足 $C_{ij} = A_i \times B_j$。所有下标均从 1 开始。

游戏一共会进行 q 轮，在每一轮游戏中，会事先给出 4 个参数 l_1, r_1, l_2, r_2，满足 $1 \le l_1 \le r_1 \le n, 1 \le l_2 \le r_2 \le m$。

游戏中，小 L 先选择一个 $l_1 \sim r_1$ 之间的下标 x，然后小 Q 选择一个 $l_2 \sim r_2$ 之间的下标 y。定义这一轮游戏中二人的得分是 C_{xy}。

小 L 的目标是使得这个得分尽可能大，小 Q 的目标是使得这个得分尽可能小。同时两人都是足够聪明的玩家，每次都会采用最优的策略。

请问：按照二人的最优策略，每轮游戏的得分分别是多少？

【输入格式】

从文件 ***game.in*** 中读入数据。

第一行输入 3 个正整数 n, m, q，分别表示数组 A，数组 B 的长度和游戏轮数。

第二行：n 个整数，表示 A_i，分别表示数组 A 的元素。

第三行：m 个整数，表示 B_i，分别表示数组 B 的元素。

接下来 q 行，每行 4 个正整数，表示这一次游戏的 l_1, r_1, l_2, r_2。

【输出格式】

输出到文件 ***game.out*** 中。

输出共 q 行，每行一个整数，分别表示每一轮游戏中，小 L 和小 Q 在最优策略下的得分。

【样例 1 输入】

```
3 2 2
0 1 -2
-3 4
1 3 1 2
2 3 2 2
```

【样例 1 输出】

```
0
4
```

【样例 1 解释】

这组数据中，矩阵 C 如下：

```
0 0
-3 4
6 -8
```

在第一轮游戏中，无论小 L 选取的是 $x = 2$ 还是 $x = 3$，小 Q 都有办法选择某个 y 使得最终的得分为负数。因此小 L 选择 $x = 1$ 是最优的，因为这样得分一定为 0。

而在第二轮游戏中，由于小 L 可以选 $x = 2$，小 Q 只能选 $y = 2$，如此得分为 4。

【样例 2 输入】

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

【样例 2 输出】

```
0
-2
3
2
-1
```

【样例 3】

见选手目录下的 ***game/game3.in*** 与 ***game/game3.ans***。

【样例 4】

见选手目录下的 ***game/game4.in*** 与 ***game/game4.ans***。

【数据范围】

对于所有数据，$1 \le n, m, q \le 10^5, -10^9 \le A_i, B_i \le 10^9$。对于每轮游戏而言，$1 \le l_1 \le r_1 \le n, 1 \le l_2 \le r_2 \le m$。

测试点编号	$n, m, q \le$	特殊条件
1	200	$1, 2$
2		1
3		2
$4 \sim 5$		无
6	1000	$1, 2$
$7 \sim 8$		1
$9 \sim 10$		2
$11 \sim 12$		无
13	10^5	$1, 2$
$14 \sim 15$		1
$16 \sim 17$		2
$18 \sim 20$		无

其中，特殊性质 1 为：保证 $A_i, B_i > 0$。

特殊性质 2 为：保证对于每轮游戏而言，要么 $l_1 = r_1$，要么 $l_2 = r_2$。

星战（galaxy）

【题目描述】

在这一轮的星际战争中，我方在宇宙中建立了 n 个据点，以 m 个单向虫洞连接。我们把终点为据点 u 的所有虫洞归为据点 u 的虫洞。

战火纷飞之中这些虫洞很难长久存在，敌人的打击随时可能到来。这些打击中的有效打击可以分为两类：

1. 敌人会摧毁某个虫洞，这会使它连接的两个据点无法再通过这个虫洞直接到达，但这样的打击无法摧毁它连接的两个据点。
2. 敌人会摧毁某个据点，由于虫洞的主要技术集中在出口处，这会导致该据点的所有还未被摧毁的虫洞被一同摧毁。而从这个据点出发的虫洞则**不会摧毁**。

注意：摧毁只会导致虫洞不可用，而不会消除它的存在。

为了抗击敌人并维护各部队和各据点之间的联系，我方发展出了两种特种部队负责修复虫洞：

- A 型特种部队则可以将某个特定的虫洞修复。
- B 型特种部队可以将某据点的所有损坏的虫洞修复。

考虑到敌人打击的特点，我方并未在据点上储备过多的战略物资。因此只要这个据点的某一条虫洞被修复，处于可用状态，那么这个据点也是可用的。

我方掌握了一种苛刻的空间特性，利用这一特性我方战舰可以沿着虫洞瞬移到敌方阵营，实现精确打击。

为了把握发动反攻的最佳时机，指挥部必须关注战场上的所有变化，为了寻找一个能够进行反攻的时刻。总指挥认为：

- 如果从我方的任何据点出发，在选择了合适的路线的前提下，可以进行无限次的虫洞穿梭（可以多次经过同一据点或同一虫洞），那么这个据点就可以**实现反击**。
- 为了使虫洞穿梭的过程连续，尽量减少战舰在据点切换虫洞时的质能损耗，当且仅当**只有一个从该据点出发的虫洞可用**时，这个据点可以**实现连续穿梭**。
- 如果我方所有据点都可以**实现反击**，也都可以**实现连续穿梭**，那么这个时刻就是一个绝佳的**反攻**时刻。

总司令为你下达命令，要求你根据战场上实时反馈的信息，迅速告诉他当前的时刻是否能够进行一次**反攻**。

【输入格式】

从文件 ***galaxy.in*** 中读入数据。

输入的第一行包含两个正整数 n, m。

接下来 m 行每行两个数 u, v，表示一个从据点 u 出发到据点 v 的虫洞。保证 $u \neq v$，保证不会有两条相同的虫洞。初始时所有的虫洞和据点都是完好的。

接下来一行一个正整数 q 表示询问个数。

接下来 q 行每行表示一次询问或操作。首先读入一个正整数 t 表示指令类型：

- 若 $t = 1$，接下来两个整数 u, v 表示敌人摧毁了从据点 u 出发到据点 v 的虫洞。保证该虫洞存在且未被摧毁。
- 若 $t = 2$，接下来一个整数 u 表示敌人摧毁了据点 u。如果该据点的虫洞已全部被摧毁，那么这次袭击不会有任何效果。
- 若 $t = 3$，接下来两个整数 u, v 表示我方修复了从据点 u 出发到据点 v 的虫洞。保证该虫洞存在且被摧毁。
- 若 $t = 4$，接下来一个整数 u 表示我方修复了据点 u。如果该据点没有被摧毁的虫洞，那么这次修复不会有任何效果。

在每次指令执行之后，你需要判断能否进行一次反攻。如果能则输出 YES 否则输出 NO。

【输出格式】

输出到文件 ***galaxy.out*** 中。

输出一共 q 行。对于每个指令，输出这个指令执行后能否进行反攻。

【样例 1 输入】

```
3 6
2 3
2 1
1 2
1 3
3 1
3 2
11
1 3 2
1 2 3
1 1 3
1 1 2
3 1 3
3 3 2
2 3
1 3 1
3 1 3
4 2
```

```
19 1 3 2
```

【样例 1 输出】

```
1  NO
2  NO
3  YES
4  NO
5  YES
6  NO
7  NO
8  NO
9  YES
10 NO
11 NO
```

【样例 1 解释】

虫洞状态可以参考下面的图片，图中的边表示存在且未被摧毁的虫洞：

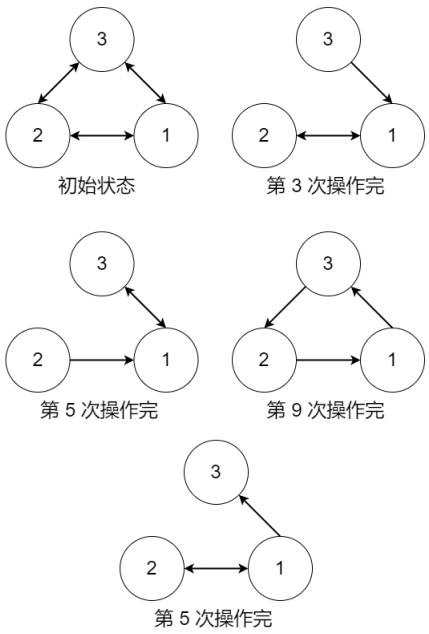

图 1: 样例 1 解释

【样例 2】

见选手目录下的 ***galaxy/galaxy2.in*** 与 ***galaxy/galaxy2.ans***。

【样例 3】

见选手目录下的 ***galaxy/galaxy3.in*** 与 ***galaxy/galaxy3.ans***。

【样例 4】

见选手目录下的 ***galaxy/galaxy4.in*** 与 ***galaxy/galaxy4.ans***。

【数据范围】

对于所有数据保证：$1 \le n \le 5 \times 10^5, 1 \le m \le 5 \times 10^5, 1 \le q \le 5 \times 10^5$。

测试点	n	m	q	特殊限制
1,2,3	≤ 10	≤ 20	≤ 50	无
4,5,6,7,8	$\le 10^3$	$\le 10^4$	$\le 10^3$	
9,10	$\le 5 \times 10^5$	$\le 5 \times 10^5$	$\le 5 \times 10^5$	保证没有 $t = 2$ 和 $t = 4$ 的情况
11,12				保证没有 $t = 4$ 的情况
13,14,15,16	$\le 10^5$			无
17,18,19,20	$\le 5 \times 10^5$			

数据传输（transmit）

【题目描述】

小 C 正在设计计算机网络中的路由系统。

测试用的网络总共有 n 台主机，依次编号为 $1 \sim n$。这 n 台主机之间由 $n-1$ 根网线连接，第 i 条网线连接个主机 a_i 和 b_i。保证任意两台主机可以通过有限根网线直接或者间接地相连。受制于信息发送的功率，主机 a 能够直接将信息传输给主机 b 当且仅当两个主机在可以通过不超过 k 根网线直接或者间接的相连。

在计算机网络中，数据的传输往往需要通过若干次转发。假定小 C 需要将数据从主机 a 传输到主机 $b(a \neq b)$,则其会选择出若干台用于传输的主机 $c_1 = a, c_2, \cdots, c_{m-1}, c_m = b$，并按照如下规则转发：对于所有的 $1 \leq i < m$, 主机 c_i 将信息直接发送给 c_{i+1}。

每台主机处理信息都需要一定的时间，第 i 台主机处理信息需要 v_i 单位的时间。数据在网络中的传输非常迅速，因此传输的时间可以忽略不计。据此，上述传输过程花费的时间为 $\sum_{i=1}^{m} v_{c_i}$。

现在总共有 q 次数据发送请求，第 i 次请求会从主机 s_i 发送数据到主机 t_i。小 C 想要知道，对于每一次请求至少需要花费多少单位时间才能完成传输。

【输入格式】

从文件 ***transmit.in*** 中读入数据。

输入的第一行包含三个正整数 n, Q, k，分别表示网络主机个数，请求个数，传输参数。数据保证 $1 \leq n \leq 2 \times 10^5, 1 \leq Q \leq 2 \times 10^5, 1 \leq k \leq 3$。

输入的第二行包含 n 个正整数，第 i 个正整数表示 v_i，保证 $1 \leq v_i \leq 10^9$。

接下来 $n-1$ 行，第 i 行包含两个正整数 a_i, b_i，表示一条连接主机 a_i, b_i 的网线。保证 $1 \leq a_i, b_i \leq n$。

接下来 Q 行，第 i 行包含两个正整数 s_i, t_i，表示一次从主机 s_i 发送数据到主机 t_i 的请求。保证 $1 \leq s_i, t_i \leq n, s_i \neq t_i$。

【输出格式】

输出到文件 ***transmit.out*** 中。

Q 行，每行一个正整数，表示第 i 次请求在传输的时候至少需要花费多少单位的时间。

【样例 1 输入】

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

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

【样例 1 输出】

```
12
12
3
```

【样例 1 解释】

对于第一组请求，由于主机 $4,7$ 之间需要至少 4 根网线才能连接，因此数据无法在两台主机之间直接传输，其至少需要一次转发；我们让其在主机 1 进行一次转发，不难发现主机 1 和主机 $4,7$ 之间都只需要两根网线即可连接，且主机 1 的数据处理时间仅为 1，为所有主机中最小，因此最少传输的时间为 $4+1+7=12$。

对于第三组请求，由于主机 $1,2$ 之间只需要 1 根网线就能连接，因此数据直接传输就是最优解，最少传输的时间为 $1+2=3$。

【样例 2】

见选手目录下的 *transmit/transmit2.in* 与 *transmit/transmit2.ans*。
该样例满足测试点 2 的限制。

【样例 3】

见选手目录下的 *transmit/transmit3.in* 与 *transmit/transmit3.ans*。
该样例满足测试点 3 的限制。

【样例 4】

见选手目录下的 *transmit/transmit4.in* 与 *transmit/transmit4.ans*。
该样例满足测试点 20 的限制。

【数据范围】

对于所有的测试数据，满足 $1 \le n \le 2 \times 10^5, 1 \le Q \le 2 \times 10^5, 1 \le k \le 3, 1 \le a_i, b_i \le n, 1 \le s_i, t_i \le n, s_i \ne t_i$。

测试点	n	Q	k	特殊性质
1	≤ 10	≤ 10	$=2$	是
2			$=3$	
3	≤ 200	≤ 200	$=2$	
4,5			$=3$	
6,7	$\le 2,000$	$\le 2,000$	$=1$	否
8,9			$=2$	
10,11			$=3$	
12,13	$\le 2 \times 10^5$	$\le 2 \times 10^5$	$=1$	
14	$\le 5 \times 10^4$	$\le 5 \times 10^4$	$=2$	是
15,16	$\le 10^5$	$\le 10^5$		
17,18,19	$\le 2 \times 10^5$	$\le 2 \times 10^5$		否
20	$\le 5 \times 10^4$	$\le 5 \times 10^4$	$=3$	是
21,22	$\le 10^5$	$\le 10^5$		
23,24,25	$\le 2 \times 10^5$	$\le 2 \times 10^5$		否

特殊性质：保证 $a_i = i + 1$，而 b_i 则从 $1, 2, \ldots, i$ 中等概率选取。