

# 8.29 练习赛

时间：2024 年 8 月 29 日 08:00 ~ 12:00

题目名称	千紫万红	狭窄回廊	崩碎之歌
题目类型	传统型	传统型	传统型
目录	game	labyrinth	shattersong
可执行文件名	game	labyrinth	shattersong
输入文件名	game.in	labyrinth.in	shattersong.in
输出文件名	game.out	labyrinth.out	shattersong.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	20	10
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	game.cpp	labyrinth.cpp	shattersong.cpp
-----------	----------	---------------	-----------------

编译选项

对于 C++ 语言	-lm -std=c++14 -O2 -Wl,--stack=2147483647
-----------	---

注意事项与提醒（请选手务必仔细阅读）

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照具体要求。
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. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 只提供 Windows 格式附加样例文件。
8. 评测在 Windows 下进行，各语言的编译器版本以其为准。

## 千紫万红 (game)

### 【题目描述】

小 M 和小 B 正在进行一场游戏。

给定一张有  $n$  个点和  $m$  条边的无向图，边有权值，点有颜色。一开始，每个点都是白色的。小 M 先手，小 M 和小 B 依次操作，每次可以选择一个白色的点，并将其染色。小 M 可以将点染成红色，小 B 可以将点染成紫色。

当图中不存在白色的点时，染色结束。当一条边的两端被染上相同颜色时，对应的玩家会获得等于其边权的得分。（红色则小 M 得分，紫色则小 B 得分）

很显然，作为先手的小 M 有绝对优势。因此，二人能否胜利的评判标准在于两人的得分差，即小 M 的得分减去小 B 的得分。小 M 希望这个值尽可能大，小 B 希望这个值尽可能小。

她们想知道，如果双方都按照最优策略选择，最终的得分差有多大？

### 【输入格式】

从文件 `game.in` 中读入数据。

本题有多测。

第一行一个整数  $T$ ，表示数据组数。

对于每组数据：

- 第一行两个整数  $n, m$ 。
- 接下来  $m$  行，每行三个整数  $u, v, w$ ，表示一条边的两个端点和边的权值。

### 【输出格式】

输出到文件 `game.out` 中。

共  $T$  行，每行一个整数，表示对应数据的答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 3
```

**【样例 1 解释】**

小 M 选择 1 号点与 3 号点，小 B 选择剩下的点。  
 小 M 获得了第 2 条边和第 4 条边，得分为  $4 + 2 = 6$ 。  
 小 B 获得了第 5 条边，得分为 3。  
 此时得分差为 3。

**【样例 2】**

见选手目录下的 *game/game2.in* 与 *game/game2.ans*。  
 该样例满足测试点 1 的限制。

**【样例 3】**

见选手目录下的 *game/game3.in* 与 *game/game3.ans*。  
 该样例满足测试点 2 的限制。

**【样例 4】**

见选手目录下的 *game/game4.in* 与 *game/game4.ans*。  
 该样例满足测试点 12 的限制。

**【样例 5】**

见选手目录下的 *game/game5.in* 与 *game/game5.ans*。  
 该样例满足测试点 20 的限制。

**【数据范围】**

测试点编号	$n$	$m$	特殊性质
1	$\leq 12$	$\leq 40$	无
2	$\leq 2 \times 10^5$	$\leq 3 \times 10^5$	A
3 ~ 12	$\leq 3 \times 10^3$	$\leq 5 \times 10^4$	无
13 ~ 20	$\leq 2 \times 10^5$	$\leq 3 \times 10^5$	无

特殊性质 A:  $u_i = 1$ 。

对于 100% 的数据,  $T \leq 4$ ,  $2 \leq n \leq 2 \times 10^5$ ,  $1 \leq m \leq 10^6$ ,  $0 \leq w_i \leq 10^9$ 。

原图可能不连通, 可能存在重边和自环。

## 狭窄回廊 (labyrinth)

### 【题目描述】

小 B 获得了一张泛黄的地图，地图标出了一个地下迷宫的结构。迷宫有  $n$  个房间和  $n - 1$  条无向边，形成一个树形结构，其中一个未知节点存有宝藏。

小 B 还有一个寻宝仪。将地图读入到寻宝仪中后，小 B 位于迷宫的一个房间时，寻宝仪会指向能走到宝藏房间方向的边（即小 B 的房间到宝藏房间的唯一路径的第一条边）。

小 B 可以依靠寻宝仪轻松找到宝藏房间，但迷宫的入口和连接两个房间的走廊早已坍塌，没有足够小 B 通过的空间。小 B 只能用传送魔法访问迷宫的房间。小 B 每次使用传送魔法，可以传送到地图上指定编号的房间，但会消耗 1 点能量。

小 B 想知道他至少需要消耗多少能量才能保证找到宝藏。注意不计小 B 最后从宝藏房间回家消耗的能量，但需要计入小 B 一开始从家传送到某个房间消耗的能量。小 B 在找到宝藏前不能提前回家。

### 【输入格式】

从文件 *labyrinth.in* 中读入数据。

本题有多测。

第一行一个整数  $T$ ，表示数据组数。

对于每组数据：

- 第一行一个整数  $n$ 。
- 接下来  $n - 1$  行，每行三个整数  $u, v$ ，表示一条边。

### 【输出格式】

输出到文件 *labyrinth.out* 中。

共  $T$  行，每行一个整数，表示对应数据的答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 3
```

**【样例 1 解释】**

树是一个链。第一次使用魔法访问 3 号节点，然后根据寻宝仪传送到 2 或 4，最后传送到 1 或 5，至多需要传送 3 次。

**【样例 2】**

见选手目录下的 *labyrinth/labyrinth2.in* 与 *labyrinth/labyrinth2.ans*。  
该样例满足测试点 2 的限制。

**【样例 3】**

见选手目录下的 *labyrinth/labyrinth3.in* 与 *labyrinth/labyrinth3.ans*。  
该样例满足测试点 4 的限制。

**【样例 4】**

见选手目录下的 *labyrinth/labyrinth4.in* 与 *labyrinth/labyrinth4.ans*。  
该样例满足测试点 20 的限制。

**【数据范围】**

测试点编号	$n$	特殊性质
1 ~ 2	$\leq 20$	无
3 ~ 4	$\leq 2 \times 10^5$	A
5 ~ 20	$\leq 2 \times 10^5$	无

特殊性质 A：树是一条链。

对于 100% 的数据， $T \leq 7$ ,  $1 \leq n \leq 2 \times 10^5$ 。

## 崩碎之歌 (shattersong)

### 【题目描述】

给你一个无向带权图  $G = (V, E)$ ，给出如下定义：

令  $W_{(u,v)}$  表示边  $(u, v)$  的边权。

定义  $C_i$  的导出子图为  $A(C_i)$ ， $A(C_i) = (C_i, \{(u, v) | u \in C_i, v \in C_j, (u, v) \in E\})$ 。

对于一个图  $G$ ，定义  $M(G)$  为对该图最小生成树的最大边权。

现在，将  $V$  分为若干个点集  $C_1, C_2 \dots C_k$ ，满足两两无交，并集为  $V$ ，且对任意一个  $i$ ，满足  $A(C_i)$  连通，则称其为一个分割。 $k$  为该分割的度。

对于一个分割，定义连接强度函数  $D(C_i, C_j) = \min_{u \in C_i, v \in C_j, (u,v) \in E} W_{(u,v)}$ 。

若满足  $u \in C_i, v \in C_j, (u, v) \in E$  的点不存在，则  $D(C_i, C_j) = +\infty$ 。

给出一个数组  $Z$ ，定义一个分割是半完美的，当且仅当对于任意一组  $i$  和  $j$ ，有  $D(C_i, C_j) > \min(M(A(C_i)) + Z_{|C_i|}, M(C_j) + Z_{|C_j|})$ 。

定义一个分割是完美的，当且仅当它是一个半完美分割，且对于任意一个  $i$ ， $A(C_i)$  不存在一个度大于 1 的半完美分割。

请构造一种完美分割的方案。

### 【输入格式】

从文件 *shattersong.in* 中读入数据。

第一行一个整数  $n, m$ 。

第二行  $n$  个整数，表示  $Z$ 。

接下来  $m$  行，每行三个整数  $u, v, w$ ，表示一条边的两个端点和边的权值。

### 【输出格式】

输出到文件 *shattersong.out* 中。

第一行一个整数  $k$ ，表示分割的度。

接下来输出  $k$  行，每行先输出  $|C_i|$ ，然后输出  $|C_i|$  个数，表示  $C_i$  中的点。

### 【样例 1 输入】

```

1 5 6
2 3 3 2 2 1
3 1 2 3
4 1 3 5
5 1 4 6
6 2 4 10
7 2 5 5
8 4 5 8

```

**【样例 1 输出】**

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

**【样例 1 解释】**

将点集分为  $\{1, 2\}, \{3\}, \{4\}, \{5\}$  这 4 个集合，这是一个完美分割。

**【样例 2】**

见选手目录下的 *shattersong/shattersong2.in* 与 *shattersong/shattersong2.ans*。  
该样例满足测试点 1 的限制。

**【样例 3】**

见选手目录下的 *shattersong/shattersong3.in* 与 *shattersong/shattersong3.ans*。  
该样例满足测试点 3 的限制。

**【样例 4】**

见选手目录下的 *shattersong/shattersong4.in* 与 *shattersong/shattersong4.ans*。  
该样例满足测试点 6 的限制。

**【样例 5】**

见选手目录下的 *shattersong/shattersong5.in* 与 *shattersong/shattersong5.ans*。  
该样例满足测试点 8 的限制。

**【样例 6】**

见选手目录下的 *shattersong/shattersong6.in* 与 *shattersong/shattersong6.ans*。  
该样例满足测试点 10 的限制。

**【数据范围】**

测试点编号	$n$	$m$	特殊性质
1	$\leq 2$	$\leq 1$	无
2 ~ 3	$\leq 10$	$\leq 45$	无
4 ~ 6	$\leq 500$	$\leq 10^5$	无
7 ~ 8	$\leq 10^5$	$\leq 5 \times 10^5$	A
9 ~ 10	$\leq 10^5$	$\leq 5 \times 10^5$	无

特殊性质 A:  $Z_i \leq Z_{i+1}$ 。

对于 100% 的数据,  $2 \leq n \leq 10^5$ ,  $1 \leq m \leq 10^5$ ,  $0 \leq W_{(u,v)}$ ,  $Z_i \leq 10^9$ 。

不保证图联通。保证图无自环和重边。