

2023 年广东省重点中学信息学邀请赛 (GDKOI 2023)

普及组 第二试

2023 年 3 月 12 日

注意事项

1. 严格按照题目所要求的格式进行输入、输出，否则严重影响得分。
2. 题目测试数据有严格的时间限制，超时不得分。
3. 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。
4. 输入文件格式不用判错；输入输出文件名均已给定，不用键盘输入。
5. 评测环境为 NOI 系列活动标准竞赛环境，编译器版本为 `g++ 9.4.0`。
6. 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）。
7. 对于 C++ 选手，64 位整数输入输出格式为 `%lld`。
8. 选手提交的程序源文件必须不大于 100KB。
9. 对于 C++ 语言的编译选项为 `-O2 -std=c++14`

试题名称	交换机	城市建设	海星	Himitsu
提交文件名	<code>switch.cpp</code>	<code>build.cpp</code>	<code>starfish.cpp</code>	<code>himitsu.cpp</code>
输入文件名	<code>switch.in</code>	<code>build.in</code>	<code>starfish.in</code>	<code>himitsu.in</code>
输出文件名	<code>switch.out</code>	<code>build.out</code>	<code>starfish.out</code>	<code>himitsu.out</code>
时间限制	1 秒	1 秒	1 秒	2 秒
空间限制	512 MB	512 MB	512 MB	256 MB
满分	100	100	100	100

第一题 交换机

提交文件: switch.cpp
 输入文件: switch.in
 输出文件: switch.out
 时间空间限制: 1 秒, 512 MB

Moon 正在准备计算机网络的开学考, 复习到了有关交换机的知识。交换机具有令人惊奇的特性那就是它的表是自动、动态和自治地建立的, 即没有来自网络管理员或来自配置协议的任何干预。换句话说, 交换机是自学习的。这种能力大概是通过如下方式实现的:

- 1) 交换机表初始为空。
- 2) 对于在每个接口接收到的每个数据帧, 该交换机在其表中存储数据帧的源 MAC 地址。
- 3) 如果在一段时间 (称为老化期) 后, 交换机没有接收到含有某个源 MAC 地址的数据帧, 就会在表中删除这个源 MAC 地址。

现在给出了某天中一个交换机所有收到的数据帧 (每一个数据帧包含一个源 MAC 地址, 以及该数据帧的到达时间), 请你帮助 Moon 算出在这一天中, 这个交换机的表最少需要多少项, 才可以使得这天所有数据帧的信息可以存储下来。**注意为了简单起见, 每个时间点先进行删除操作再进行插入操作。**

简单来说, 需要维护一个字符串集合 S , 有若干次插入操作, 每次插入操作包含一个字符串以及一个有效时间区间 (区间长度为定值), 需要求出所有时间内, 字符串集合 S 大小的最大值。

输入格式

第一行两个整数 n, k , 分别表示数据帧的个数, 以及老化期时间 (单位: 分钟)。

接下来 n 行, 每行两个字符串 s_1, s_2 , 表示数据帧包含的源地址帧, 以及到达时间, 其中 s_1 是一个 MAC 地址, 包含 12 个 16 进制数, s_2 的格式为 $a:b$, 其中 $0 \leq a \leq 23, 0 \leq b \leq 59$, 表示该数据帧在 a 时 b 分钟 0 秒到达交换机, 并且 a, b 不足 10 的数字前面要补充一个零。

输出格式

一行, 一个整数, 代表答案。

样例数据

switch.in	switch.out
4 10 0123456789ABCDEF 00:10 0000000000ABCDEF 08:11 0123456789ABCDEF 00:15 0000000000ABCDEF 00:11	2
3 60 0123456789ABCDEF 13:00 0000000000000000 14:00 0123456789ABCDEF 12:30	1
见/example/switch/下的 switch1.in	见/example/switch/下的 switch1.out

样例解释

样例 1 中, 时刻 00:11 分表项中有 2 个源 MAC 地址 0123456789ABC 和 0000000000ABCDEF, 所以至少需要表项大小至少为 2, 可以证明这是最少需要的表项大小。

数据范围

对于所有的数据, 有 $1 \leq n \leq 10^5, 1 \leq k \leq 1440$;

对于 50% 的数据, 有 $1 \leq n \leq 500$ 。

第二题 城市建设

提交文件: build.cpp
 输入文件: build.in
 输出文件: build.out
 时间空间限制: 1 秒, 512 MB

现在有 N 个城市分别位于 $1, 2, \dots, N$ 号节点上。作为城市的管理者小明希望花最小的代价把这 N 个城市联通起来。联通城市的花费主要分为下面两种。

第一种：建设道路，在 a 号城市和 b 号城市之间建设道路将会花费 $|a - b|$ 的代价；

第二种：设置管理城市，对于一个城市 X 可以使用 C 的代价将其升级为管理城市；

在保证每条道路至少联结一个管理城市，以及非管理城市只能连接一条边的前提下，小明想知道他最少能用多少的代价使所有城市联通。

输入格式

第一行一个正整数 N ，表示城市数量；

第二行一个整数 C ，表示设置管理城市的代价。

输出格式

一行，一个整数，表示小明使城市联通的最小代价。

样例数据

build.in	build.out
6 3	12
1000 1000	32561
100000 10000	10099799

样例解释

以 2, 5 分别为管理城市，并联结 $(1, 2), (2, 3), (2, 5), (4, 5), (5, 6)$ 共五条边时取得最小代价；也可以以 3 或 4 作为管理城市，并联结其余所有点，此时也取得最小代价。

数据范围

对于 20% 的数据， $1 \leq N \leq 20$

对于 40% 的数据， $1 \leq N \leq 10^3$ ；

对于另外 20% 的数据， $1 \leq N \leq 10^5, 0 \leq C \leq 10^4$ ；

对于 100% 的数据， $1 \leq N \leq 10^9, 0 \leq C \leq 10^9$ 。

第三题 海星

提交文件: starfish.cpp
 输入文件: starfish.in
 输出文件: starfish.out
 时间空间限制: 1 秒, 512 MB

小明想去海底抓海星, 海底是一个树状的结构, 而海星就藏在里面。

给定一棵 n 个点的树, 分别标号为 $1, 2, 3, \dots, n$, 且对应的点权记为 p_i 。

定义海星为其中的花子图, 不妨设其中心节点标号为 O , 边缘节点标号为 a_1, a_2, \dots, a_t (其中边缘节点必定直接与中心节点相连, 同时花子图至少由一个中心节点和一个边缘节点构成)。此时记海星的价值为 $|p_O - \sum p_{a_i}|$ 。

小明想知道他最多能抓到价值总和为多少的海星。(可以同时抓很多只海星, 但是任意两个海星之间点的交集必须为空)

补充定义:

花图: 直径小于等于三的联通图, 其中度数最大的节点为中心节点, 其余节点为边缘节点 (可知任意的边缘节点度数为 1)

示例: 最小的花图, 是 $(G, V) = (\{1, 2\}, \{(1, 2)\})$, 仅由两个节点和联结它们的一条边构成

输入格式

第一行, 一个正整数 n , 表示树的大小;

第 2 行, n 个整数, 表示 p_i ;

第 3 ~ $n + 1$ 行, 每行 2 个整数 a, b , 分别表示 a 号节点与 b 号节点之间有一条边。

输出格式

一行, 一个整数, 表示小明能抓到的最大海星的价值总和。

样例数据

starfish.in	starfish.out
5 -2 -3 4 -5 1 1 2 1 3 2 4 2 5	10
见/example/starfish/下	见/example/starfish/下

样例解释

一个合法的方案是, 小明抓了两个海星, 第一个海星的中心节点为 1, 边缘节点为 3, 价值为 6; 第二个海星的中心节点为 2, 边缘节点为 5, 价值为 4, 此时得到最大总价值 10。

数据范围

- (1) 对于 10% 的数据，保证数据形成一个花图
- (2) 对于 20% 的数据，保证数据形成一条链。
- (3) 对于 20% 的数据，保证数据相邻节点乘积恒负。
- (4) 对于 20% 的数据，保证数据形成一颗以 1 为根节点的二叉树。
- (5) 对于 30% 的数据，无特殊限制。

对于所有数据 $1 \leq a, b \leq n \leq 10^5$, $-10^9 \leq p_i \leq 10^8$ 。

下发样例中编号为 $i, i + 5$ 的数据对应序号为 i 的限制条件. $i \in \{1, 2, 3, 4, 5\}$

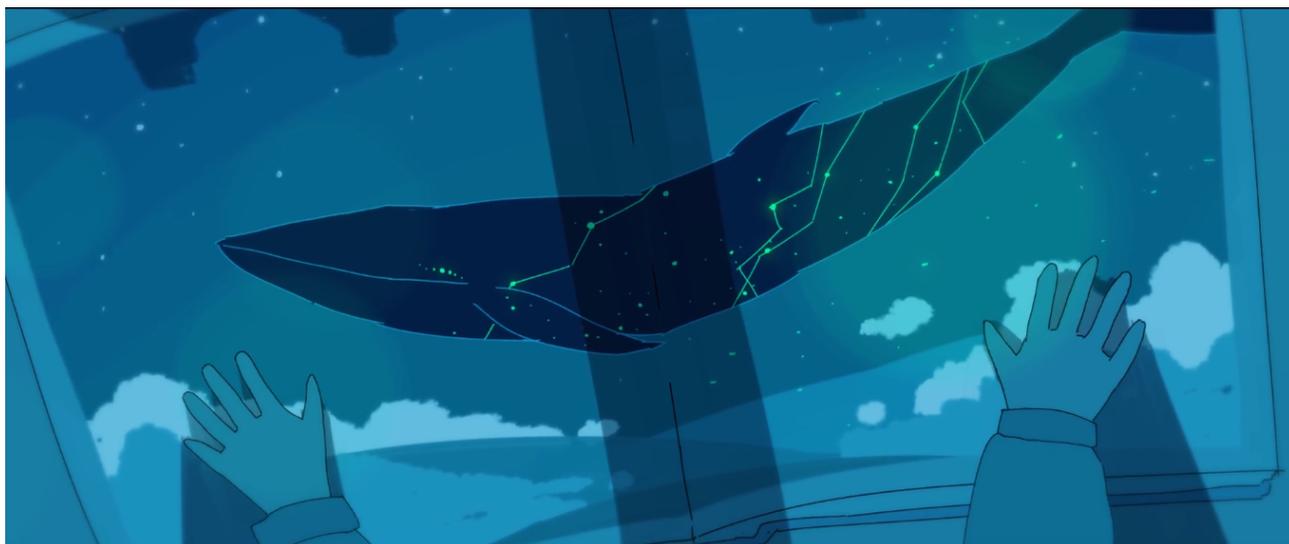
第四题 Himitsu

提交文件: himitsu.cpp
输入文件: himitsu.in
输出文件: himitsu.out
时间空间限制: 2 秒, 256 MB



Lily 忘不掉小时候临别前与 Nana 一起出逃的夜晚，两人为了探索宇宙的真相而沿着铁路轨道前行，望见从天际线涌出的大鱼浮在夜空里。从那以后 Lily 便未曾停止过钻研宇宙的秘密。

秘密啊，总是会有那么一两个的。以浮起的大鱼为典型，Lily 所在的世界发生了 n 件关于宇宙的重大事件。阴谋诡计，完全犯罪，宇宙的暗物质，事件与事件之间看似互不相干，但 Lily 每天阅读着关于宇宙的书籍，从书籍最后一页总结出 m 段解读，每段解读都能将其中某两件事件联系起来。



大人们希望 Lily 揭露部分已经得到的解读，使所有发生的事件都能被串联起来。但是 Lily 清楚，如果这些解读暴露，班里大概有一半的同学会开始计划着出逃——秘密的揭露都是需要 Lily 付出代价的，假设每一段解读的揭露需要付出的代价可以被量化，第 i 段解读的揭露需要付出的代价为整数值 w_i 。Lily 需要付出的总代价为，在确保解读们能将所有事件联系起来的前提下，Lily 所揭露的所有解读的代价之和。

Lily 知道，其中有 k 段解读是关于世界真理的关键解读，是承载着 70 亿的秘密，这些解读被透露的数

量将直接决定着人类延续的可能性。人们议论纷纷，说着或许只有这两个孩子才能拯救世界，70 亿人所凝聚的正义将 2 人的自由剥夺，迫切地想要知道 Lily 给出的答案。

你要知道，对于 0 到 k 中所有整数 i ，Lily 透露了一部分解读，能将所有事件联系起来，且其中正好有 i 个关键解读的前提下，Lily 需要付出的最小总代价。



“我不明白啊”，Lily 这样的哭了。她会坚守着秘密，按照约定等待着与 Nana 再会的那一刻。

输入格式

第一行有三个整数 n, m, k ，分别表示事件数量，解读总数和关键解读的数量。

紧接着 m 行每行有三个整数 u, v, w ，分别表示一种能将事件 u 与事件 v 联系起来的解读，但是这解读的揭露需要付出 w 的代价。

这 m 行中前 k 行表示 Lily 所知道的关键解读。

输出格式

$k + 1$ 行每行有一个整数 ans 。其中第 i 行整数 ans_i 表示揭露正好 $i - 1$ 个关键解读的。

数据保证在所有关键解读都不揭露的情况下，剩余的解读能将所有事件联系在一起。

样例数据

himitsu.in	himitsu.out
5 8 2	21
3 4 6	19
2 1 6	20
1 4 8	
1 2 10	
2 3 4	
3 4 5	
4 5 4	
2 4 6	

数据范围

$$1 \leq u, v \leq n, 1 \leq w \leq 10^9$$

对于 30% 的数据满足 $1 \leq n \leq 100, 1 \leq m \leq 400, 1 \leq k \leq 5$

对于另外 30% 的数据满足 $1 \leq n \leq 10000, 1 \leq m \leq 1000000, 1 \leq k \leq 20$

对于剩下 40% 的数据满足 $1 \leq n \leq 10000, 1 \leq m \leq 1000000, 1 \leq k \leq 50$

赛题被启用的时候，小小的行星上已经有 80 亿人了