

实外 CCF CSP2022-S模拟赛

(提高组：第5场)

时间：2022 年 10 月 15 日 8:00 ~ 12:00

题目引用：NOIP2018

题目名称	旅行	铺设道路	货币系统	赛道修建
题目类型	传统型	传统型	传统型	传统型
目录	travel	road	money	track
可执行文件名	travel	road	money	track
输入文件名	travel.in	road.in	money.in	track.in
输出文件名	travel.out	road.out	money.out	track.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512MB	512MB
子任务数目	25	10	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	travel.cpp	road.cpp	money.cpp	track.cpp
对于 C 语言	travel.c	road.c	money.c	track.c
对于 Pascal 语言	travel.pas	road.pas	money.pas	track.pas

编译选项

对于 C++ 语言	-O2
对于 C 语言	-O2
对于 Pascal 语言	-O2

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. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 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。
8. 只提供 Linux 格式附加样例文件。
9. 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。

1. 旅行

(travel.cpp/c/pas)

【问题描述】

小 Y 是一个爱好旅行的 OIer。她来到 X 国，打算将各个城市都玩一遍。

小 Y 了解到，X 国的 n 个城市之间有 m 条双向道路。每条双向道路连接两个城市。不存在两条连接同一对城市的道路，也不存在一条连接一个城市 and 它本身的道路。并且，从任意一个城市出发，通过这些道路都可以到达任意一个其他城市。小 Y 只能通过这些道路从一个城市前往另一个城市。

小 Y 的旅行方案是这样的：任意选定一个城市作为起点，然后从起点开始，每次可以选择一条与当前城市相连的道路，走向一个**没有去过**的城市，或者沿着**第一次**访问该城市时经过的道路后退到上一个城市。当小 Y 回到起点时，她可以选择结束这次旅行或继续旅行。需要注意的是，小 Y 要求在旅行方案中，每个城市都被访问到。

为了让自己的旅行更有意义，小 Y 决定在每到达一个新的城市（包括起点）时，将它的编号记录下来。她知道这样会形成一个长度为 n 的序列。她希望这个序列的字典序最小，你能帮帮她吗？

对于两个长度均为 n 的序列 A 和 B，当且仅当存在一个正整数 x ，满足以下条件时，我们说序列 A 的字典序小于 B。

- 对于任意正整数 $1 \leq i < x$ ，序列 A 的第 i 个元素 A_i 和序列 B 的第 i 个元素 B_i 相同。
- 序列 A 的第 x 个元素的值小于序列 B 的第 x 个元素的值。

【输入格式】

输入文件名为 travel.in。

输入文件共 $m + 1$ 行。第一行包含两个整数 $n, m (m \leq n)$ ，中间用一个空格分隔。

接下来 m 行，每行包含两个整数 $u, v (1 \leq u, v \leq n)$ ，表示编号为 u 和 v 的城市之间有一条道路，两个整数之间用一个空格分隔。

【输出格式】

输出文件名为 travel.out。

输出文件包含一行， n 个整数，表示字典序最小的序列。相邻两个整数之间用一个空格分隔。

【输入输出样例 1】

travel.in	travel.out
6 5 1 3 2 3 2 5 3 4 4 6	1 3 2 5 4 6

见选手目录下的 travel/travell.in 和 travel/travell.ans。

【输入输出样例 2】

travel.in	travel.out
6 6 1 3 2 3 2 5 3 4 4 5 4 6	1 3 2 4 5 6

见选手目录下的 travel/travel2.in 和 travel/travel2.ans。

【输入输出样例 3】

见选手目录下的 travel/travel3.in 和 travel/travel3.ans。

这组样例满足 $m = n - 1$ 。

【输入输出样例 4】

见选手目录下的 travel/travel4.in 和 travel/travel4.ans。

这组样例满足 $m = n$ 。

【数据规模与约定】

对于 100% 的数据和所有样例， $1 \leq n \leq 5000$ 且 $m = n - 1$ 或 $m = n$ 。

对于不同的测试点，我们约定数据的规模如下：

测试点编号	$n =$	$m =$	特殊性质
1, 2, 3	10	$n - 1$	无
4, 5	100		无
6, 7, 8	1000		每个城市最多与两个城市相连
9, 10	1000		无
11, 12, 13	5000		每个城市最多与三个城市相连
14, 15	5000		无
16, 17	10	n	无
18, 19	100		无
20, 21, 22	1000		每个城市最多与两个城市相连
23, 24, 25	5000		无

2. 铺设道路

(road.cpp/c/pas)

【问题描述】

春春是一名道路工程师，负责铺设一条长度为 n 的道路。

铺设道路的主要工作是填平下陷的地表。整段道路可以看作是 n 块首尾相连的区域，一开始，第 i 块区域下陷的深度为 d_i 。

春春每天可以选择一段连续区间 $[L, R]$ ，填充这段区间中的每块区域，让其下陷深度减少 1。在选择区间时，需要保证，区间内的每块区域在填充前下陷深度均不为 0。

春春希望你能帮他设计一种方案，可以在最短的时间内将整段道路的下陷深度都变为 0。

【输入格式】

输入文件名为 road.in。

输入文件包含两行，第一行包含一个整数 n ，表示道路的长度。

第二行包含 n 个整数，相邻两数间用一个空格隔开，第 i 个整数为 d_i 。

【输出格式】

输出文件名为 road.out。

输出文件仅包含一个整数，即最少需要多少天才能完成任务。

【输入输出样例 1】

road.in	road.out
6 4 3 2 5 3 5	9

见选手目录下的 road/road1.in 和 road/road1.ans。

【样例解释】

一种可行的最佳方案是，依次选择：

$[1,6]$ 、 $[1,6]$ 、 $[1,2]$ 、 $[1,1]$ 、 $[4,6]$ 、 $[4,4]$ 、 $[4,4]$ 、 $[6,6]$ 、 $[6,6]$ 。

【输入输出样例 2】

见选手目录下的 road/road2.in 和 road/road2.ans。

【数据规模与约定】

对于 30% 的数据， $1 \leq n \leq 10$ ；

对于 70% 的数据， $1 \leq n \leq 1000$ ；

对于 100% 的数据， $1 \leq n \leq 100000$ ， $0 \leq d_i \leq 10000$ 。

3. 货币系统

(money.cpp/c/pas)

【问题描述】

在网友的国度中共有 n 种不同面额的货币，第 i 种货币的面额为 $a[i]$ ，你可以假设每一种货币都有无穷多张。为了方便，我们把货币种数为 n 、面额数组为 $a[1..n]$ 的货币系统记作 (n, a) 。

在一个完善的货币系统中，每一个非负整数的金额 x 都应该可以被表示出，即对每一个非负整数 x ，都存在 n 个非负整数 $t[i]$ 满足 $a[i] \times t[i]$ 的和为 x 。然而，在网友的国度中，**货币系统可能是不完善的**，即可能存在金额 x 不能被该货币系统表示出。例如在货币系统 $n=3$ ， $a=[2, 5, 9]$ 中，金额 $1, 3$ 就无法被表示出来。

两个货币系统 (n, a) 和 (m, b) 是等价的，当且仅当对于任意非负整数 x ，它要么均可以被两个货币系统表出，要么不能被其中任何一个表出。

现在网友们打算简化一下货币系统。他们希望找到一个货币系统 (m, b) ，满足 (m, b) 与原来的货币系统 (n, a) 等价，且 m 尽可能的小。他们希望你来协助完成这个艰巨的任务：找到最小的 m 。

【输入格式】

输入文件名为 money.in。

输入文件的第一行包含一个整数 T ，表示数据的组数。接下来按照如下格式分别给出 T 组数据。

每组数据的第一行包含一个正整数 n 。接下来一行包含 n 个由空格隔开的正整数 $a[i]$ 。

【输出格式】

输出文件名为 money.out。

输出文件共有 T 行，对于每组数据，输出一行一个正整数，表示所有与 (n, a) 等价的货币系统 (m, b) 中，最小的 m 。

【输入输出样例 1】

money.in	money.out
2	2
4	5
3 19 10 6	
5	
11 29 13 19 17	

见选手目录下的 money/money1.in 和 money/money1.ans。

【输入输出样例 1 说明】

在第一组数据中，货币系统 $(2, [3, 10])$ 和给出的货币系统 (n, a) 等价，并可以验证不存在 $m < 2$ 的等价的货币系统，因此答案为 2。

在第二组数据中，可以验证不存在 $m < n$ 的等价的货币系统，因此答案为 5。

【输入输出样例 2】

见选手目录下的 money/money2.in 和 money/money2.ans。

【数据规模与约定】

测试点	n	a_i	测试点	n	a_i
1	$= 2$	≤ 1000	11	≤ 13	≤ 16
2			12		
3			13		
4	$= 3$		14	≤ 25	≤ 40
5			15		
6			16		
7	$= 4$		17	≤ 100	≤ 25000
8			18		
9	$= 5$		19		
10			20		

对于 100% 的数据，满足 $1 \leq T \leq 20$, $n, a[i] \geq 1$ 。

4. 赛道修建

(`track.cpp/c/pas`)

【问题描述】

C 城将要举办一系列的赛车比赛。在比赛前，需要在城内修建 m 条赛道。

C 城一共有 n 个路口，这些路口编号为 $1, 2, \dots, n$ ，有 $n - 1$ 条适合于修建赛道的双向通行的道路，每条道路连接着两个路口。其中，第 i 条道路连接的两个路口编号为 a_i 和 b_i ，该道路的长度为 l_i 。借助这 $n - 1$ 条道路，从任何一个路口出发都能到达其他所有的路口。

一条赛道是一组互不相同的道路 e_1, e_2, \dots, e_k ，满足可以从某个路口出发，依次经过道路 e_1, e_2, \dots, e_k （每条道路经过一次，不允许调头）到达另一个路口。一条赛道的长度等于经过的各道路的长度之和。为保证安全，要求每条道路至多被一条赛道经过。

目前赛道修建的方案尚未确定。你的任务是设计一种赛道修建的方案，使得修建的 m 条赛道中长度最小的赛道长度最大（即 m 条赛道中最短赛道的长度尽可能大）。

【输入格式】

输入文件名为 `track.in`。

输入文件第一行包含两个由空格分隔的正整数 n, m ，分别表示路口数及需要修建的赛道数。

接下来 $n - 1$ 行，第 i 行包含三个正整数 a_i, b_i, l_i ，表示第 i 条适合于修建赛道的道路连接的两个路口编号及道路长度。保证任意两个路口均可通过这 $n - 1$ 条道路相互到达。每行中相邻两数之间均由一个空格分隔。

【输出格式】

输出文件名为 `track.out`。

输出共一行，包含一个整数，表示长度最小的赛道长度的最大值。

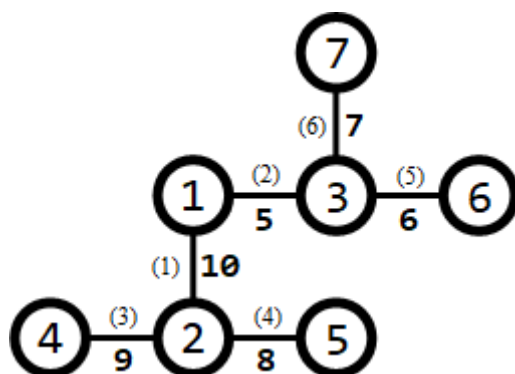
【输入输出样例 1】

<code>track.in</code>	<code>track.out</code>
7 1 1 2 10 1 3 5 2 4 9 2 5 8 3 6 6 3 7 7	31

见选手目录下的 `track/track1.in` 与 `track/track1.ans`。

【输入输出样例 1 说明】

所有路口及适合于修建赛道的道路如下图所示：



道路旁括号内的数字表示道路的编号，非括号内的数字表示道路长度。

需要修建 1 条赛道。可以修建经过第 3,1,2,6 条道路的赛道（从路口 4 到路口 7），则该赛道的长度为 $9 + 10 + 5 + 7 = 31$ ，为所有方案中的最大值。

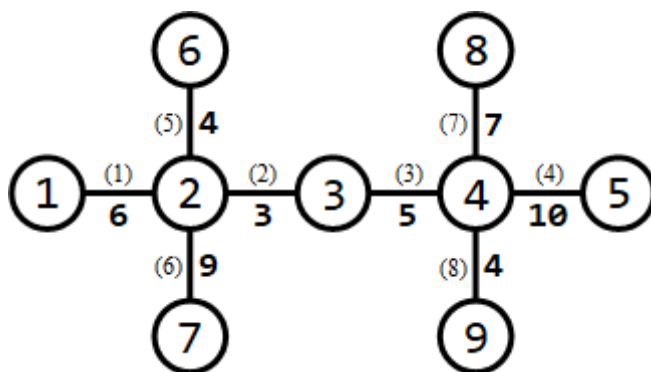
【输入输出样例 2】

track.in	track.out
9 3 1 2 6 2 3 3 3 4 5 4 5 10 6 2 4 7 2 9 8 4 7 9 4 4	15

见选手目录下的 track/track2.in 与 track/track2.ans。

【输入输出样例 2 说明】

所有路口及适合于修建赛道的道路如下图所示：



需要修建 3 条赛道。可以修建如下 3 条赛道：

1. 经过第 1,6 条道路的赛道（从路口 1 到路口 7），长度为 $6 + 9 = 15$ ；
2. 经过第 5,2,3,8 条道路的赛道（从路口 6 到路口 9），长度为 $4 + 3 + 5 + 4 = 16$ ；
3. 经过第 7,4 条道路的赛道（从路口 8 到路口 5），长度为 $7 + 10 = 17$ 。

长度最小的赛道长度为 15，为所有方案中的最大值。

【输入输出样例 3】

见选手目录下的 track/track3.in 与 track/track3.ans。

【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	n	m	$a_i = 1$	$b_i = a_i + 1$	分支不超过 3				
1	≤ 5	$= 1$	否	否	是				
2	≤ 10	$\leq n - 1$		是		是			
3	≤ 15		$= 1$	否	否	否			
4	$\leq 1,000$	否		是					
5	$\leq 30,000$	是		否					
6		否							
7		是							
8	$\leq 50,000$	否	是		是				
9	$\leq 1,000$								
10	$\leq 30,000$		否						
11	$\leq 50,000$								
12	≤ 50								
13									
14	≤ 200								
15									
16	$\leq 1,000$				否				
17									
18	$\leq 30,000$								
19									
20	$\leq 50,000$								

其中，“分支不超过 3”的含义为：每个路口至多有 3 条道路与其相连。

对于所有的数据, $2 \leq n \leq 50,000, 1 \leq m \leq n - 1, 1 \leq a_i, b_i \leq n, 1 \leq l_i \leq 10,000$ 。