

2018-9-29 NOIP 训练-Day1

一、题目概览

中文题目名称	聪明的小偷	无名	服务
英文题目名称	thief	noname	service
可执行文件名	thief.exe	noname.exe	service.exe
输入文件名	thief.in	noname.in	service.in
输出文件名	thief.out	noname.out	service.out
每个测试点时限	1s	1s	3s
测试点数目	10	10	20
每个测试点分值	10	10	5
比较方式	全文比较	全文比较	全文比较
题目类型	传统	传统	传统
运行内存上限	256M	256M	256M

二、提交源程序文件名

对于 Pascal 语言	thief.pas	noname.pas	service.pas
对于 C 语言	thief.c	noname.c	service.c
对于 C++ 语言	thief.cpp	noname.cpp	service.cpp

注意事项：

1. 今天是提高组 Day1 信心赛，题目难度相对较低，大家尽管屠。
2. 有关评测环境以及编译命令的问题，请咨询你们的老师。

——By zld3794955

1.聪明的小偷

(thief.pas/c/cpp)

【问题描述】

从前有一个收藏家收藏了许多相同的硬币,并且将它们放在了 n 个排成一排的口袋里,每个口袋里都装了一定数量的硬币。

这些硬币价值不菲,自然引起不少人觊觎(ji, 4)觊(yu, 2),于是收藏家每天都会来检查一次这 n 个口袋,首先他会先检查每个口袋是不是都有硬币,之后他会计算出第 1 个和第 2 个口袋的硬币数量之和,第 2 个与第 3 个口袋的硬币数量和,如此直到第 $n-1$ 个与第 n 个口袋的硬币数量之和,得到 $n-1$ 个数的序列。

如果收藏家发现某个口袋没有硬币,或者他计算得到的序列较上一天相比有变动,那么收藏家就知道肯定有人动了他的硬币。

有一个聪明的小偷,他想在收藏家不知道的情况下偷走一些硬币,为此,他不仅可以偷偷地从某个口袋中拿出一些硬币,也可以将硬币在口袋间任意移动,现在他想知道对于给定的 n 个口袋及对应的硬币数量,他最多能拿多少枚硬币。

小偷是很聪明的,他早就算出来啦,但是他想考考作为徒弟的你……

【输入格式】

第一行一个正整数 n ,表示口袋的个数。

接下来一行 n 个正整数,第 i 个正整数表示第 i 个口袋里装的硬币数量。

【输出格式】

一行一个正整数,表示答案。

【输入输出样例】

thief.in	thief.out
3 1 2 3	0
4 2 4 6 8	0
5 2 3 4 5 6	1

【样例解释】

第一个样例中,小偷无法在收藏家不知道的情况下移动硬币。

第二个样例中,尽管小偷可以移动硬币,比如移动成 3 3 7 7,但是他仍然无法拿出硬币。

第三个样例中,小偷将 1,5 口袋的一个硬币移动到 2,4 口袋中,并在口袋 3 中拿出一个硬币即可。

【数据规模与约定】

对于 50%的数据,有 $2 \leq n \leq 20$,每个口袋中硬币数量 ≤ 20 。

对于 100%的数据,有 $2 \leq n \leq 999$,每个口袋中硬币数量 ≤ 10000 且为正整数。

2.无名

(noname.pas/c/cpp)

【问题描述】

这道题实在不知道该取什么名字比较好，于是就取了这个名字。

给定一个长度为 n 的正整数序列，你的任务就是求出至少需要修改序列中的多少个数才能使得该数列成为一个严格(即不允许相等)单调递增的正整数序列，对序列中的任意一个数，你都可以将其修改为任意的正整数。

【输入格式】

每个测试点第一行为一个正整数 T ，表示该测试点内的数据组数，你需要对该测试点内的 T 组数据都分别给出正确的答案才能获得该测试点的分数。

接下来 T 组数据，每组数据第一行一个正整数 n ，表示序列长度，接下来一行 n 个正整数描述这个序列。

【输出格式】

对于每一组测试数据输出一行一个非负整数表示答案。

【输入输出样例】

noname.in	noname.out
4	2
3	1
1 1 2	0
3	2
2 2 3	
4	
1 2 3 4	
4	
2 3 3 4	

【样例解释】

第一组数据中，由于正整数的限制，只能修改序列中第 2、3 个数。

第二组数据中，将第一个数修改为 1 即可。

第三组数据已经符合条件，不需要修改。

第四组数据中，由于正整数的限制，只修改一个数并不能完成任务，而需要同时修改前两个数或者后两个数。

【数据规模与约定】

对于 30% 的数据，有 $1 \leq n \leq 10$ 。

对于 80% 的数据，有 $1 \leq n, T \leq 50$ ，且输入序列中每个数均不超过 50。

对于 100% 的数据，有 $1 \leq n \leq 50000$ ， $1 \leq T \leq 5000$ ，输入序列中每个数均为不超过 1000000000 (10^9) 的正整数，且每个测试点中 T 组数据对应的 n 值总和不超过 500000。

3.服务

(service.pas/c/cpp)

【问题描述】

一家公司为它在各地的用户提供服务，公司有三名负责这项工作的员工，分别编号为 1, 2, 3，服务的地点有 n 个，分别编号为 1, 2, 3, ..., n ，把从编号为 p 的服务地点直接到达编号为 q 的服务地点所需的移动费用记为 $C(p, q)$ ，显然 $C(p, p)=0$ （停留在原地不需要费用），但不保证对任意 p, q 均有 $C(p, q)=C(q, p)$ 。

初始时员工 1 在地点 1，员工 2 在地点 2，员工 3 在地点 3，现在公司依次收到了 L 个服务请求，每个请求需要一名员工赶到其指定的地点进行服务，员工可以选择直达，也可以选择经过若干个服务地点中转，特别地，如果指派的员工已在当前请求所在地，则该请求不需要任何移动费用即可被处理。

出于公平起见，所有请求必须按顺序处理，这意味着即使一名员工在赶往当前请求的途中经过之后的请求所在的地点，他也不可以先处理之后的请求，但是公司不限制每位员工赶往请求地点的路线，也允许一个服务地点有多名员工。

你的任务就是对于这 L 个请求，找到一个服务方案（即对每个请求分配合适的员工去服务以及规划移动路线），使得三名员工提供服务的总移动费用最小。

【输入格式】

每个测试点第一行为一个正整数 T ，表示该测试点内的数据组数，你需要对该测试点内的 T 组数据都分别给出正确的答案才能获得该测试点的分数。

接下来 T 组数据，每组数据第一行两个正整数 n 和 L ，含义如题目所述，接下来 n 行，每行 n 个非负整数，其中第 p 行第 q 个数表示 $C(p, q)$ ，最后一行包含一行 L 个正整数，依次描述每个请求指定的地点。

【输出格式】

对每组数据输出一行一个非负整数，表示最小的总移动费用。

【输入输出样例】

service.in	service.out
2	0
3 6	5
0 1 2	
3 0 4	
5 6 0	
1 2 3 1 2 3	
5 9	
0 1 1 1 1	
1 0 2 3 2	
1 1 0 4 1	
2 1 5 0 1	
4 2 3 4 0	
4 2 4 1 5 4 3 2 1	

【样例解释】

第一组测试数据中，所有可能的地点都有员工，因此不需要进行任何移动。

第二组测试数据中，三位员工初始在1 2 3三个地方，下面给出其中一种达到最小总移动费用的方案：

请求地点	指派员工	移动路线	员工所在地	当前总代价
(初始状态)			1 2 3	0
4	1	1->4	4 2 3	0+1=1
2	2	2	4 2 3	1+0=0
4	1	4	4 2 3	1+0=0
1	2	2->1	4 1 3	1+1=2
5	2	1->5	4 5 3	2+1=3
4	1	4	4 5 3	3+0=3
3	3	3	4 5 3	3+0=3
2	1	4->2	2 5 3	3+1=4
1	3	3->1	2 5 1	4+1=5

【数据规模与约定】

每个测试点 5 分，各个测试点数据范围如下：

测试点编号	n	L
1-3	$n \leq 10$	$L \leq 10$
4-6	$n \leq 40$	$L \leq 10$
7-10	$n \leq 40$	$L \leq 100$
11-14	$n \leq 100$	$L \leq 100$
15-17	$n \leq 100$	$L \leq 500$
18-20	$n \leq 200$	$L \leq 500$

对于所有的测试点，均有数据组数 $T \leq 5$ ，地点数 $n \geq 3$ ，给定的 C 矩阵主对角线上的数全部为 0，且输入数据中所有的数均为不超过 2000 的非负整数。