凌逸教育暑假集训

LingYi Regular Round 8

一、 题目概况

编号	中文题目名称	时间限制	空间限制
A	阿笠加法（Easy Version）	1s	256 MB
B	阿笠加法（Hard Version）	1s	256 MB
C	目暮警官的查案分组	1s	256 MB
D	怪盗准则	1s	256 MB
E	汉诺塔推理	2s	512 MB

二、 注意事项

1. 本次测试采用标准输入输出，无需文件读入读出。
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. 评测时采用的机器配置为：CPU AMD Athlon(tm) II x2 240 processor，2.8GHz，内存 4G，上述时限以此配置为准。

A.阿笠加法（Easy Version）

(A.cpp/c/pas)

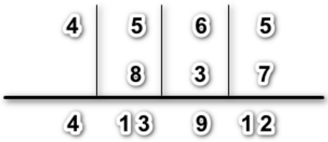
时间限制：1s　　空间限制：256 MB

【问题描述】

阿笠博士发明了一个新的加法运算规则，并取名为“阿笠加法”，准备在周末出游路上给少年侦探团做智力问答。“阿笠加法”的运算规则如下：

阿笠博士会提供两个正整数 A，B。需要将两个数的最低位对齐后对应位的数字相加得到和，并将和按位顺序相连的值输出。

例如，两个正整数分别为 4565 和 837 时，运算结果就是 413912，运算过程如图所示。



阿笠博士为了防止自己在侦探团面前出现计算错误，请帮助他先算出正确答案。

【输入格式】

第一行，两个正整数 A，B。

【输出格式】

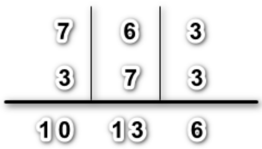
一行，一个正整数，表示 A，B 两数经过“阿笠加法”后得到的值。

【输入输出样例】

输入 1	**输出 1**
763 373	10136

【样例解释】

对于第一组样例： 结果为 10136，运算过程见下图。



【数据范围】

$100 \le A, B < 10^3$;

B. 阿笠加法（Hard Version）

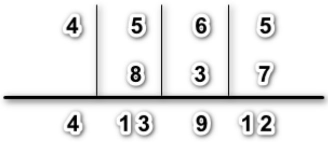
(B.cpp/c/pas)
时间限制：1s　　空间限制：256 MB

【问题描述】

阿笠博士发明了一个新的加法运算规则，并取名为“阿笠加法”，准备在周末出游路上给少年侦探团做智力问答。“阿笠加法”的运算规则如下：

阿笠博士会提供两个正整数 A，B。需要将两个数的最低位对齐后对应位的数字相加得到和，并将和按位顺序相连的值输出。

例如，两个正整数分别为 4565 和 837 时，运算结果就是 413912，运算过程如图所示。



阿笠博士为了防止自己在侦探团面前出现计算错误，请帮助他先算出正确答案。

【输入格式】

第一行，两个正整数 A，B。

【输出格式】

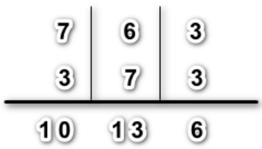
一行，一个正整数，表示 A，B 两数经过“阿笠加法”后得到的值。

【输入输出样例】

输入 1	输出 1
763 373	10136

【样例解释】

对于第一组样例：　结果为 10136，运算过程见下图。



【数据范围】

$1 \le A, B \le 10^{1000}$；

C. 目暮警官的查案分组

(C.cpp/c/pas)

时间限制：1s　　空间限制：256 MB

【问题描述】

以东京为中心发生了连环杀人案件，各县刑警都参加了联合调查，其中目暮十三作为警视厅刑事部资历最为丰富的搜查课警部，成为了本次联合调查的指挥官，负责分配人手查案。

本次联合调查一共有n名刑警参加，第i名刑警的资历为a_i。目暮警官在分组时为了避免出现刑警资历过低，小组能力过差，无法顺利完成任务的情况，决定在分组时增加一定的限制：小组的人数与该小组资历最低的刑警的资历的乘积不能低于限制值x。

对于每个刑警，只能被安排进一个队伍；有些刑警可能不会被安排进队伍。

请帮助目暮警官判断他最多能将这些刑警分成几个组。

【输入格式】

第一行，一个正整数t，表示t组测试数据。

接下去$2 \cdot t$ 行，每两行为一组。对于其中第一行，两个正整数n，x。n表示刑警的个数，x表示限制值。对于其中第二行，n 各正整数$a_1, a_2 ..., a_n$。a_i表示第i名刑警的资历值。

【输出格式】

t行，每行一个整数，表示对于每组测试数据，最多能分组的个数。

【输入输出样例】

输入 1	输出 1
3 5 10 7 11 2 9 5 4 8 2 4 2 3 4 11 1 3 3 7	2 1 0

【样例解释】

对于第一组数据：最多可以分成两组，可以分为将 1，2 号刑警分为一组，4，5 号刑警分为一组。也可以将 2 号刑警分为一组，1，4，5 号刑警分为一组，分组方案不唯一。

对于第二组数据：最多可以分成一组，可能的分组方案为 1，2，3，4 号刑警分成一组。

对于第三组数据：不存在满足限制条件的分组。

【数据范围】

$1 \leq t \leq 10^3$;

$1 \leq n \leq 10^5$;

$1 \leq x \leq 10^9$;

$1 \leq a_i \leq 10^9$;

对于所有的t组数据，n的和不超过 10^5;

D. 怪盗准则

(D.cpp/c/pas)

时间限制：1s　　空间限制：256 MB

【问题描述】

铃木财团将在东京举办一场大型珠宝展，怪盗基德潜入了铃木财团的收藏室准备偷取这批价值连城的珠宝。收藏室内按顺序放置了n件珠宝，第i号位置存放的珠宝价值为a_i，且这些珠宝中不存在价值相同的珠宝。

怪盗基德偷窃珠宝必须要遵从怪盗准则，每条怪盗准则的时间消耗不同，方式如下：

准则一：绝不多拿！基德可以选择花费x时间，拿走连续的k件珠宝。

准则二：绝不贪心！基德可以选择花费y时间，选择连续的两件珠宝，并拿走其中价值更小的一件。

例如，珠宝价值依次为{2,3,7,8,11,5,4}，且可以连续拿走3件珠宝，即$k = 3$时。基德可以选择先根据准则二选择连续的两件珠宝，其价值为8和11，拿走其中价值为8的珠宝，剩余珠宝价值依次为{2,3,7,11,5,4}。然后，基德也可以选择根据准则一连续拿掉3件珠宝，以拿掉{7,11,5}为例，剩余珠宝价值依次为{2,3,4}。

怪盗基德对于他不感兴趣的珠宝，坚决不会下手。收藏室内珠宝价值依次为$\{a_1, a_2, \dots, a_n\}$，怪盗基德对其中的m件珠宝并不感兴趣，在他盗窃后剩余的珠宝价值依次为$\{b_1, b_2, \dots, b_m\}$。请问他最少需要花费多少时间可以带走所有他想要的珠宝。

【输入格式】

第一行，两个正整数n，m，分别表示收藏室内所有珠宝和基德不感兴趣的珠宝的个数。

第二行，三个正整数x，k，y，分别表示准则一需要花费的时间，准则一可以连续拿走的珠宝的件数和准则二需要花费的时间。

第三行，n个正整数$a_1, \dots, a_n$，表示收藏室内n件珠宝依次摆放后各自的价值。

第四行，m个正整数$b_1, \dots, b_m$，表示收藏室被盗窃后剩余m件珠宝依次摆放各自的价值。

【输出格式】

一行，一个整数，表示基德带走所有他想要的珠宝最少需要花费的时间，如果无法遵循怪盗准则带走基德想要的珠宝，请输出“-1”。

【输入输出样例】

输入 1	输出 1
6 2 4 2 3 3 6 2 1 5 4 6 4	10

输入 2	输出 2
3 2 4 2 1 1 2 3 3 2	-1

输入 3	输出 3
3 3 4 2 1 1 2 3 1 2 3	0

【样例解释】

对于第一组样例:

第一步，遵循准则二，选出{3,6}，拿走{3}的珠宝，剩余珠宝为{6,2,1,5,4}，消耗 3 时间;

第二步，遵循准则二，选出{2,1}，拿走{1}的珠宝，剩余珠宝为{6,2,5,4}，消耗 3 时间;

第三步，遵循准则一，拿走连续两件珠宝{2,5}，剩余珠宝为{6,4}，消耗 4 时间;

总计消耗 3+3+4 = 10 时间。

对于第二组样例，不存在基德行窃后残留珠宝价值顺序为{3,2}的情况，输出“-1”。

对于第三组样例，基德没有想要偷窃的珠宝，所以消耗时间为 0。

【数据范围】

$1 \le n, m \le 2 \cdot 10^5$;

$1 \le x, y \le 10^9$;

$1 \le k \le n$;

$1 \le a_i \le n$;

$1 \le b_i \le n$;

E. 汉诺塔推理

(F.cpp/c/pas)

时间限制：2s　　空间限制：512 MB

【问题描述】

侦探的必备素养就是推理的逻辑性！柯南为了锻炼自己的推理逻辑，给自己定制了一个汉诺塔训练。

柯南手上有n个大小不一的但形状相同的圆盘，以及m根用来堆放圆盘的柱子，其编号为$1 \sim n$。对于第i号圆盘，其直径为i。开始时，这些圆盘分堆被堆在m根柱子上，每根柱子上至少有一个圆盘，且每根柱子上的圆盘都是从大到小依次从柱子底部向上堆的。

柯南每次可以选择两根柱子（柱子上至少堆有一个圆盘），从一个柱子i的顶部开始移动若干圆盘到另一柱子j上，且这若干圆盘必须满足直径都比j号柱子顶端圆盘的直径小。

例如，开始时 1 号柱子和 2 号柱子上圆盘从下往上排列顺序分别为{6,4,2,1}和{8,7,5,3}。柯南可以从 1 号柱子上移动 1 个圆盘到 2 号柱子，移动后变为{6,4,2}和{8,7,5,3,1}；也可以从 1 号柱子上移动 2 个圆盘到 2 号柱子，移动后变为{6,4}和{8,7,5,3,2,1}。

柯南需要进行若干次移动，最终将所有的圆盘移动到一个柱子上。为了锻炼的逻辑推理能力，柯南要求自己每次都移动最少的次数将所有圆盘移动到一个柱子上。为了增加难度，柯南在每次决定好初始的圆盘位置，并推理出最少的移动次数后，对圆盘任意两根非空的柱子进行$m-1$次合并，并将每次合并结束后的若干堆圆盘的最少移动次数推理出来。

对于第i次合并的柱子a_i，b_i，需要将a_i和b_i号柱子上所有的圆盘从下往上，从大到小全部合并到a_i号柱子上。

柯南为了验证自己的准确性，希望你告诉他每次他进行推理时的正确答案。

【输入格式】

第一行，两个正整数n，m，表示有n个圆盘，m根柱子。

第二行，n个正整数$t_1, t_2, \dots, t_n$，表示圆盘的初始状态，其中t_i表示编号为i的圆盘被放置在t_i号柱子上。

接下去$m-1$行，每行两个正整数a_i，b_i，表示第i次合并是将a_i和b_i柱子上所有的盘子全部合并到a_i号柱子上。

【输出格式】

输出m行，第一行表示初始状态下最少需要移动的次数，第i行表示第$i-1$次合并后最少需要移动的次数。

【输入输出样例】

输入 1	**输出 1**
6 3 1 3 3 2 1 3 3 1 2 3	4 2 0

【样例解释】

初始状态柱子上圆盘的摆放为：{5,1},{4},{6,3,2}
最少移动的顺序为：
第一次移动：{5,1},{4},{6,3,2}→{5,1},{4,3,2},{6}
第二次移动：{5,1},{4,3,2},{6}→{5},{4,3,2,1},{6}
第三次移动：{5},{4,3,2,1},{6}→{5,4,3,2,1},{},{6}
第四次移动：{5,4,3,2,1},{},{6}→{},{},{6,5,4,3,2,1}
故输出 4。

对于第 1 次合并，将 1 号 3 号柱子上的圆盘合并到 3 号圆盘上，摆放为：{},{4},{6,5,3,2,1}
最少移动顺序为：
第一次移动：{},{4},{6,5,3,2,1}→{},{4,3,2,1},{6,5}
第二次移动：{},{4,3,2,1},{6,5}→{},{},{6,5,4,3,2,1}
故输出 2.

对于第 2 次合并，在上一次合并的基础上，将 2 号 3 号柱子上的圆盘合并到 2 号圆盘上，摆放为：{},{6,5,4,3,2,1},{}。已经满足要求无需移动，输出 0。

【数据范围】

$2 \le m \le n \le 2 \cdot 10^5$；
$2 \le t_i \le m$；
$1 \le a_i, b_i \le m$，且$a_i \ne b_i$；