

全国信息学奥林匹克联赛(模拟赛)

提高组

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	动物园	线段计数	选择数字
英文题目名称	zoo	segment	number
可执行文件名	zoo	segment	number
输入文件	zoo.in	segment.in	number.in
输出文件	zoo.out	segment.out	number.out
每个测试点时限	2 秒	5 秒	1 秒
测试点数目	20	10	10
每个测试点分值	5	10	10
附加样例文件	有	有	有
结果比较方式	全文比较(过滤行末空格及文末回车)		
题目类型	传统	传统	传统
运行内存	128M	128M	64M

二. 提交源程序文件名

对于 C++语言	zoo.cpp	segment.cpp	number.cpp
对于 C 语言	zoo.c	segment.c	number.c
对于 pascal 语言	zoo.pas	segment.pas	number.pas

三. 编译命令(不包含任何优化开关)

对于 C++语言	g++ -o zoo zoo.cpp -lm	g++ -o segment segment.cpp -lm	g++ -o number number.cpp -lm
对于 C 语言	gcc -o zoo zoo.c -lm	gcc -o segment segment.c -lm	gcc -o number number.c -lm
对于 pascal 语言	fpc zoo.pas	fpc segment.pas	fpc number.pas

注意事项:

- 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, 程序正常结束返回值必须是 0。
- 3、在 Windows XP 的环境下进行最终评测, 请 C/C++选手使用 %I64d 或 cin/cout 来读入输出 64 位整数。

1. 动物园

(zoo. cpp/c/pas)

【问题描述】

小 Z 计划带妹子去逛动物园。动物园有 n 个区域，从 1 到 n 编号。第 i 个区域有 a_i 只动物。动物园里有 m 条道路，道路是无向的，每条道路连接两个区域，且动物园是连通的，每两个区域之间都有路径相连。

小 Z 为了哄妹子高兴，需要提前规划旅行路线。对于两个区域 p 和 q ($p \neq q$) 间的任意一条简单路径，定义其高兴度为路径上经过的区域中最少的动物数量（包括区域 p 和区域 q ）。小 Z 自然希望高兴度越高越好，定义 p 、 q ($p \neq q$) 间最大的高兴度为 p 、 q 间所有简单路径中高兴度的最大值，记作 $f(p, q)$ ，现在小 Z 想知道，对于动物园任意两点的 $f(p, q)$ 总和是多少。

【输入格式】

输入文件名为 zoo. in。

输入的第一行为两个整数 n 和 m ，第二行包含 n 个整数，表示每个区域动物的数量。

接下来的 m 行，每行包含两个数 x_i 和 y_i ($x_i \neq y_i$)，表示区域 x_i 和 y_i 之间有一条道路。

【输出格式】

输出文件名为 zoo. out。

输出包含一行，一个整数表示最大高兴度总和，即 $\sum_{p \neq q} f(p, q)$ 。

【输入输出样例】

zoo. in	zoo. out
4 5 40 20 30 40 1 2 2 3 1 3 2 4 3 4	300

【输入输出样例说明】

$$f(1,2) = 20 \quad f(1,3) = 30 \quad f(1,4) = 30$$

$$f(2,1) = 20 \quad f(2,3) = 20 \quad f(2,4) = 20$$

$$f(3,1) = 30 \quad f(3,2) = 20 \quad f(3,4) = 30$$

$$f(4,1) = 30 \quad f(4,2) = 20 \quad f(4,3) = 30$$

总和为300

【数据规模与约定】

对于 10%的测试数据，满足 $2 \leq n \leq 10, n-1 \leq m \leq 20$ 。

对于 30%的测试数据，满足 $2 \leq n \leq 200, n-1 \leq m \leq 1000$ 。

对于 50%的测试数据，满足 $2 \leq n \leq 1000, n-1 \leq m \leq 20000$ 。

对于 100%的测试数据，满足 $2 \leq n \leq 100000, n-1 \leq m \leq 1000000$,

$0 \leq a_i \leq 100000, 1 \leq x_i, y_i \leq n, (x_i \neq y_i)$ ，保证是连通图且没有重边。

2. 线段计数

(segment.cpp/c/pas)

【问题描述】

小 Z 是一个聪明的孩子，因此经常能收到礼物。

有一天，小 Z 从小 P 那里得到了长度为 $1, 2, 3 \dots$ 的线段作为礼物，聪明的小 Z 想将这些线段摆放在数轴上。第 i 次插入操作，将长度为 i 的线段放在数轴上，每一次插入操作后，他都会统计刚刚插入的线段完全覆盖了之前多少条线段。聪明的小 Z 觉得只插入线段太简单了，于是他又添加了删除的操作，对于删除操作，他都会从数轴上删掉第 k 次插入的线段（线段是相互独立的），现在请你来帮助小 Z 解决这个问题。

【输入格式】

输入文件名为 segment.in。

每个测试点包含多组测试数据。

输入的第一行为一个正整数 T ，表示测试数据的组数。

每组测试数据包含 $n + 1$ 行。

第一行包含一个正整数 n ，代表操作的次数，接下来 n 行每行包含一个操作，每个操作用两个整数 x 和 y 表示。

如果 x 是 0，表示插入操作，小 P 在 y 位置插入一条线段（对于第 i 次插入的线段，线段将被插入到 $[y, y + i]$ ，长度为 i ）。

如果 x 是 1，表示删除操作，小 P 会删除第 y 次插入操作插入的线段。

【输出格式】

输出文件名为 segment.out。

对于每组测试数据，首先输出一行‘Case #i:’（不包含引号），表示第 i 组测试数据的输出。

接着对于每次的插入操作，按题目要求输出答案。

【输入输出样例】

segment. in	segment. out
2	Case #1:
5	0
0 1	0
0 3	0
0 5	1
1 3	Case #2:
0 2	0
5	1
0 1	0
0 1	2
1 1	
0 2	
0 1	

【输入输出样例说明】

对于第一组数据：

第一次插入[1, 2]线段，没有完全覆盖任何线段，输出 0；

第二次插入[3, 5]线段，没有完全覆盖任何线段，输出 0；

第三次插入[5, 8]线段，没有完全覆盖任何线段，输出 0；

删除第三次插入的线段；

第四次插入[2, 6]线段，完全覆盖第二次插入的线段，输出 1。

【数据规模与约定】

对于 10%的测试数据，满足 $1 \leq n \leq 100$ 。

对于 30%的测试数据，满足 $1 \leq n \leq 2000$ 。

对于 70%的测试数据，满足 $1 \leq n \leq 50000$ 。

对于 100%的测试数据，满足 $1 \leq T \leq 5, 1 \leq n \leq 160000$ ；

对于插入操作满足 $|y| < 10^9$ ；

对于删除操作，保证 y 合法，每个线段最多会被删除 1 次。

3. 选择数字 (number.cpp/c/pas)

【问题描述】

给定 N 个正整数 A_1, A_2, \dots, A_N ，从中选出若干个，使它们的和为 M ，求有多少种选择方案。

【输入格式】

输入文件名为 number.in。

输入第一行包含两个整数 N 和 M 。

第二行包含 N 个正整数 A_1, A_2, \dots, A_N 。

【输出格式】

输出文件名为 number.out。

输出只有一行，一个整数表示最后的答案。

【输入输出样例】

number.in	number.out
4 4 1 1 2 2	3

【输入输出样例说明】

从 1 1 2 2 四个数中选取若干个，使其和为 4，共有 3 种选法分别为 (1, 1, 2)、(1, 1, 2)、(2, 2)。

【数据规模与约定】

对于 40% 的数据， $1 \leq N \leq 20, 1 \leq M \leq 10$

对于 100% 的数据， $1 \leq N \leq 100, 1 \leq M \leq 10000, 1 \leq A_i \leq 1000$