

CCF 全国信息学奥林匹克联赛（NOIP2020）复赛

提高组模拟

时间：2020 年 11 月 30 日 8:00–12:00

一. 题目概况

中文题目名称	删边	剪辣椒	开灯	柱形图
英文题目与子目录名	delete.cpp	chilli.cpp	light.cpp	histogram.cpp
可执行文件名	传统型	传统型	传统型	传统型
输入文件名	delete	chilli	light	histogram
输出文件名	delete	chilli	light	histogram
每个测试点时限	1秒	1秒	2秒	2.5秒
测试点数目	4	3	4	2
每个测试点分值	捆绑测试	捆绑测试	捆绑测试	捆绑测试
附加样例文件				
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	传统
运行内存限制	512MB	512MB	512MB	512MB

二. 提交源程序文件名

对于C++语言	delete.cpp	chilli.cpp	light.cpp	histogram.cpp
对于C语言	delete.c	chilli.c	light.c	histogram.c
对于pascal语言	delete.pas	chilli.pas	light.pas	histogram.pas

三. 编译命令（不包含任何优化开关）

对于C++语言	g++ -o delete delete.cpp -lm	g++ -o chilli chilli.cpp -lm	g++ -o light light.cpp -lm	g++ -o histogram histogram.cpp -lm
对于C语言	gcc -o delete delete.c -lm	gcc -o chilli chilli.c -lm	gcc -o light light.c -lm	gcc -o histogram histogram.c -lm
对于pascal语言	fpc delete.pas	fpc chilli.pas	fpc light.pas	fpc histogram.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，程序正常结束时的返回值必须是 0。
- 3、全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。

4、只提供 Linux 格式附加样例文件。

5、特别提醒：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其为准。

1. 删边

(delete.cpp/c/pas)

【题目描述】

给你一棵 n 个结点的树，每个结点有一个权值，删除一条边的费用为**该边连接的两个子树中结点权值最大值**之和。现要删除树中的所有边，删除边的顺序可以任意设定，请计算出所有方案中的最小花费。

【输入格式】

输入文件名为 delete.in。

第一行包含整数 n ，表示结点数。结点用从 1 到 n 表示。

第二行包含 n 个整数 t_i ($1 \leq t_i \leq 109$)。数字 t_i 表示结点 i 的权值。

接下来 $n-1$ 行，每行包含两个整数 x 和 y ($1 \leq x, y \leq n$)，表示结点 x 和结点 y 直接相连。

【输出格式】

输出文件名为 delete.out。

输出最小花费。

【输入输出样例 1】

delete.in	delete.out
3 1 2 3 1 2 2 3	8

【输入输出样例 2】

delete.in	delete.out
4 2 2 3 2 1 3 3 2 4 3	15

【输入输出样例 3】

delete.in	delete.out
5 5 2 3 1 4 2 1 3 1 2 4 2 5	26

【数据范围】

子任务	限制	分值
1	$1 \leq n \leq 10$	10
2	结点 i ($1 \leq i \leq n-1$) 和 $i+1$ 直接相连	10
3	$1 \leq n \leq 1000$	30
4	$1 \leq n \leq 100,000$	50

2. 剪辣椒

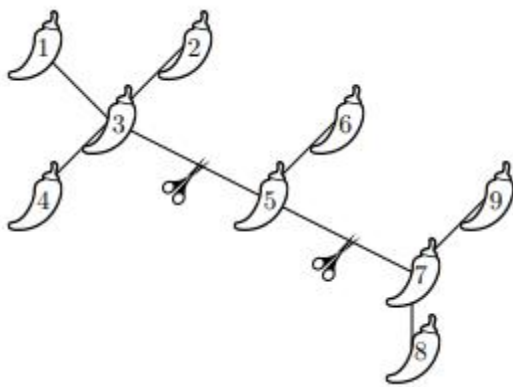
(chilli.cpp/c/pas)

【题目描述】

在花园里劳累了一上午之后，你决定用自己种的干辣椒奖励自己。

你有 n 个辣椒，这些辣椒用 $n-1$ 条绳子连接在一起，任意两个辣椒通过用若干个绳子相连，即形成一棵树。

你决定分三餐吃完这些辣椒，因此需要剪断其中两根绳子，从而得到三个组成部分，每一餐吃一个组成部分即可。



每一餐不可以太辣，所以你会寻找一个剪绳子的方法，使得最大组成部分和最小组成部分的辣椒数量差最小。计算出这个最小差值。

【输入格式】

输入文件名为 `chilli.in`。

第一行一个整数 n ，表示辣椒的数量。辣椒从 1 到 n 进行编号。

下面 $n-1$ 行每一行包含两个整数 x 和 y ($1 \leq x, y \leq n$)，表示辣椒 x 和辣椒 y 直接用一根绳子相连。

【输出格式】

输出文件名为 `chilli.out`。

输出最小差值。

【输入输出样例 1】

chilli.in	chilli.out
4	1
1 2	
2 3	
3 4	

【输入输出样例 2】

chilli.in	chilli.out
6	0
1 2	
1 3	
3 4	
3 5	
5 6	

【输入输出样例 3】

chilli.in	chilli.out
9	2
1 3	
2 3	
3 4	
3 5	
5 6	
5 7	
7 8	
7 9	

【数据范围】

子任务	限制	分值
1	$3 \leq n \leq 200$	15
2	$3 \leq n \leq 2000$	35
3	$3 \leq n \leq 200,000$	50

3. 关灯

(light.cpp/c/pas)

【题目描述】

你房间的吊灯坏了。吊灯由 n 个灯泡组成，这些灯泡由 $n-1$ 根电线连接，每根电线连接两个灯泡并且任意两个灯泡直接或通过其他灯泡相连。换句话说，吊灯是一棵树。

每个灯泡都有一个独立的改变其状态的按钮。如果灯泡关着，按一下按钮灯泡就会打开，如果灯泡开着，按一下按钮灯泡就会关闭。一开始，有些灯泡是开着的，有些是关的（有可能所有灯泡都关闭）。所有的 n 个灯泡都需要打开，这样你才不会害怕。

你将会选择一连串的灯泡，这一连串的灯泡是连续的且相邻两个灯泡是直接通过电线相连的，灯泡可以在序列中出现多次。你会按照序列的顺序走到对应的灯泡处，并按下当前灯泡的按钮，改变它的状态。

计算出最短的灯泡序列，使得按照上面的操作后所有灯泡都处于打开状态。题目保证至少有一个灯泡在开始时处于关闭。

【输入格式】

输入文件名为 light.in。

第一行包含一个整数 n ，表示灯泡的数量。灯泡用 1 到 n 来编号。

第二行包含一个长度为 n 的“0”和“1”组成的序列，描述一开始灯泡的状态。“0”表示关闭，“1”表示打开。

接下来 $n-1$ 行每行都包含两个整数 x 和 y ($1 \leq x, y \leq n$)，表示灯泡 x 和灯泡 y 直接相连。

【输出格式】

输出文件名为 light.out。

输出一个序列的最小可能长度，以使所有灯泡最终打开。可以证明这样的序列总是存在的。

【输入输出样例 1】

light.in	light.out
3 010 1 2 2 3	4

【输入输出样例 2】

light.in	light.out
5 00000 1 2	7

2 3	
2 4	
3 5	

【输入输出样例 3】

light. in	light. out
5	8
00100	
1 2	
2 3	
2 4	
3 5	

【数据范围】

对所有数据满足： $2 \leq n \leq 500,000$

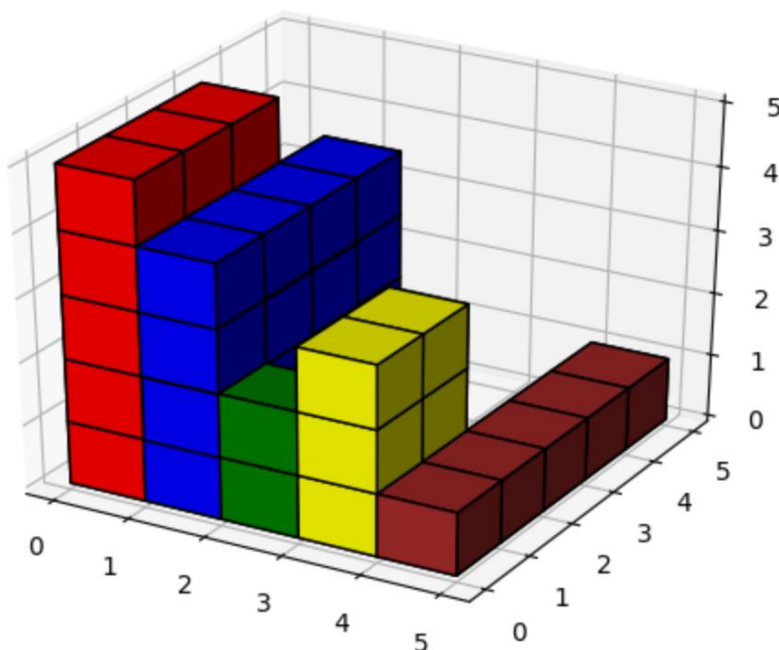
子任务	限制	分值
1	$2 \leq n \leq 100$	20
2	每个灯泡最多与两个其他灯泡直接相连	20
3	所有灯泡一开始全部处于关闭状态	20
4	无	40

4. 柱形图

(`histogram.cpp/c/pas`)

【题目描述】

给你一个三维柱形图，它由 n 个相邻的块组成。第 i 个块有 1 米宽， a_i 米高， b_i 米长。也就是说，从前面看这个块，它看起来像一个有多条高度为 a_1, a_2, \dots, a_n 的长条组成的柱形图，从顶部看像一个高度为 b_1, b_2, \dots, b_n 的柱形图。



计算可以放进这个 3D 柱形图的块的最大体积，要求这个块的边要与构成这个 3D 柱形图的块的边平行。

【输入格式】

输入文件名为 `histogram.in`。

第一行包含一个整数 n 。

接下来 n 行，每行包含两个整数 a_i 和 b_i ($1 \leq a_i, b_i \leq 10^6$)

【输出格式】

输出文件名为 `histogram.out`。

输出能放进该 3D 柱形图的块的最大体积。

【输入输出样例 1】

histogram.in	histogram.out
5 5 3 4 4 2 1 3 2 1 5	24 解释：见上图

【输入输出样例 2】

histogram.in	histogram.out
6 3 1 2 1 2 2 2 3 1 1 2 2	8

【输入输出样例 3】

histogram.in	histogram.out
5 15 19 5 6 1 13 3 7 1 2	285

【数据范围】

子任务	限制	分值
1	$1 \leq n \leq 2000$	20
2	$1 \leq n \leq 200,000$	80