

NOIP2021 模拟赛

ywy_c_asm

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	逛公园	苹果树	工厂建设	密码锁
英文题目与子目录名	park	apple	factory	psw
可执行文件名	park	apple	factory	psw
输入文件名	park.in	apple.in	factory.in	psw.in
输出文件名	park.out	apple.out	factory.out	psw.out
每个测试点时限	1s	2s	1s	1s
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
附加样例文件	无	无	无	无
结果比较方式	全文比较 (过滤行末空格和回车)			
题目类型	传统	传统	传统	传统
运行内存上限	512MB	512MB	512MB	512MB

二、提交源程序文件名

对于 C++语言	park.cpp	apple.cpp	factory.cpp	psw.cpp
----------	----------	-----------	-------------	---------

三、编译命令

对于 C++语言	g++ -o park park.cpp -lm	g++ -o apple apple.cpp -lm	g++ -o factory factory.cpp -lm	g++ -o psw psw.cpp -lm
----------	-----------------------------	-------------------------------------	---	---------------------------

四、良心提示

1. 题目并不难。
2. 部分分一定要能写则写，不要直接弃疗。

1. 逛公园

(park.cpp)

【题目描述】

3 岁小孩小小 Y 是小 Y 的儿子，现在小小 Y 要去独自一人逛公园。公园可以抽象为 n 个点 m 条边的无向图，每条边表示一条长度为 1 的连接两个地点的道路。1 号点为公园的入口，小小 Y 将从这里出发，走一条简单路径（即不会到达同一个点两次）到达一个终点，然后他将在这里玩耍，直到小 Y 过来接他为止。

为了保证小小 Y 的安全，小 Y 给他立了两条规矩：①走的总路程不能超过 3，②终点必须与 1 号点有边相连（即“在入口点旁边”）。

虽然小小 Y 清楚这样逛到的地方一定少得可怜，小小 Y 想知道总共有多少种逛公园的不同方案。

【输入格式】

输入数据第一行两个整数 n, m ，表示点数和边数。

接下来 m 行，每行两个整数 s, t ，表示一条边。保证无自环、重边。

【输出格式】

一行一个整数，表示方案数。

【输入输出样例】

输入样例	输出样例
4 5 1 2 1 3 2 4 3 4 2 3	6

【样例解释】

逛公园的全部方案为：1->2, 1->3, 1->2->3, 1->3->2, 1->2->4->3, 1->3->4->2。

【数据规模与约定】

对于 100% 的数据， $1 \leq n, m \leq 3 \times 10^5$

测试点编号	$n, m \leq$
1, 2, 3	500
4, 5, 6	10000
7, 8, 9, 10	300000

2. 苹果树

(apple.cpp)

【题目描述】

小 Y 有一棵苹果树，它可以被抽象为一棵 n 个点的有根树，每个点上结了一个重量为 w_i 的苹果。身患强迫症的小 Y 产生了 m 个想法，它们分别表示在某个点的子树中必须恰好摘 c_i 个苹果。现在小 Y 要摘总重量尽可能大的苹果，并且还要满足这些想法。

【输入格式】

输入数据第一行 3 个整数 n, m, rt ，表示点数，想法数以及树根所在点的编号。

接下来 $n-1$ 行，每行两个数 s, t ，表示一条树边。

接下来一行 n 个整数表示每个点上的苹果重量。

接下来 m 行，每行两个整数 x, y ，表示小 Y 的一个想法是在 x 的子树中恰好摘 y 个苹果。

【输出格式】

若小 Y 的想法都能被满足，输出一个整数表示能摘得的苹果的最大总重量。否则输出“naive”（不带引号）。

【输入输出样例】

输入样例	输出样例
5 2 1 1 2 1 3 2 4 2 5 1 2 3 4 5 1 3 2 2	12

【样例解释】

小 Y 可以摘 3、4、5 苹果，这样 1 的子树里摘了 3 个，2 的子树里摘了 2 个，满足想法。

【提示】

由于输入输出规模较大，建议使用较为高效的读入方式。

【数据规模与约定】

对于 100% 的数据， $1 \leq rt, x, m \leq n \leq 10^6$ ， $1 \leq w_i \leq 10^9$ ，保证没有两个想法的 x 是相同的。

测试点编号	特殊性质
1	$n \leq 20$
2, 3	$n \leq 2000$
4	$m = 0$
5	$m = 1$
6, 7, 8, 9, 10	无

3. 工厂建设

(factory.cpp)

【题目描述】

矿产资源丰富的 Byteland 大陆吸引了包括小 Y 在内的众多资本家的目光。通过一系列肮脏邪恶的手段，小 Y 成功地把 Byteland 大陆据为己有，完成了资本的原始积累。他准备在这里建设 k 个工厂，为自己创造更多的利益。

Byteland 大陆可以抽象为一棵 n 个点的树形结构，每条边都有一个长度 l_i ，**工厂必须建在度数为 1 的叶结点上（仅与一个点有边相连）**，一个点上最多只能建一个工厂。由于工厂之间要进行物资的互相运输，因此工厂两两之间的距离不能太大。定义工厂建设的总代价为 k 个工厂两两之间距离的总和。

作为被资本家雇(bo)佣(xue)的劳动者，你需要帮小 Y 求出最小的建设总代价。

【输入格式】

输入数据第一行 2 个整数 n, k ，表示点数和工厂数。

接下来 $n-1$ 行，每行 3 个整数 s, t, l ，表示一条连接 s 和 t 的长度为 l 的边。

【输出格式】

一行一个整数，表示最小的建设总代价。数据保证建设方案存在。

【输入输出样例】

输入样例 1	输出样例 1
4 3 1 2 2 1 3 3 1 4 4	18
输入样例 2	输出样例 2
4 2 1 2 2 1 3 3 1 4 4	5

【数据规模与约定】

对于 100% 的数据， $1 \leq k \leq 100$ ， $2 \leq n \leq 10^5$ ， $k \leq n$ ， $1 \leq l_i, w_i \leq 10^9$

测试点编号	特殊性质
1, 2	$n \leq 15$
3, 4	$n \leq 200$
5, 6	叶节点至少有 $n-1$ 个
7, 8, 9, 10	无

4. 密码锁

(psw.cpp)

【题目描述】

小 Y 新买了一把密码锁！往里面输入正确的密码，就会把门打开。

然而，小 Y 总是在喝酒喝得迷迷糊糊时回家输入密码开门，难免会输错几个字符。不过良心的制造商考虑到了这个问题，若输入的密码与正确密码不匹配，但它们的“编辑距离”不超过 3，那么密码锁会给予适当提示。定义两个字符串 S 和 T 的编辑距离 $d(S, T)$ 为，每次操作修改 T 某个位置上的字符，或者删除 T 的某个字符，或者在 T 的某个位置插入任意一个字符，将 T 变成 S 的最少操作次数。

现在你需要实现这个密码锁。

【输入格式】

输入数据第一行一个整数 T 表示数据组数。

接下来 T 组数据，第一行一个字符串 A 表示正确密码，第二行一个字符串 B 表示小 Y 输入的密码。

【输出格式】

对于每组数据，若 $A=B$ ，则输出 "You're logged in!"。

若 $d(A, B)=1$ ，则输出 "You almost got it. You're wrong in just one spot."。

若 $d(A, B)=2$ ，则输出 "You almost got it, but you're wrong in two spots."。

若 $d(A, B)=3$ ，则输出 "You're wrong in three spots."。

若 $d(A, B)>3$ ，说明小 Y 喝多了，输入的都是胡言乱语，则输出 "What you entered is too different from the real password."。

【输入输出样例】

输入样例	输出样例
5	You're logged in!
password	You almost got it. You're wrong in just one spot.
password	You almost got it, but you're wrong in two spots.
password	You're wrong in three spots.
passw0rd	What you entered is too different from the real password.
password	
pazzword	
password	
pazzw0rd	
password	
username	

【数据规模与约定】

对于 100% 的数据， $1 \leq |A|, |B| \leq 10^5$ ，字符串总长不超过 10^6 且其中字符的 ASCII 码值都在 33~126 之间。

测试点编号	$ A , B \leq$	总串长 \leq
1, 2, 3	100	100
4, 5	1000	10000
6, 7	100000	200000
8, 9, 10	100000	1000000