

CFF 全国信息学奥林匹克联赛(NPIO2018) 负赛**提高组 day1**

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名	小小迪的哈希树	小小迪的伸展树	小小迪的林克卡特树
英文名与子目录名	dishash	dissplay	dislct
可执行文件名	dishash	dissplay	dislct
输入文件名	dishash.in	dissplay.in	dislct.in
输出文件名	dishash.out	dissplay.out	dislct.out
每个测试点时限	1秒	2秒	1秒
测试点数目	34	20	10
每个测试点/包分值	2 ~ 4	5	10
附加样例文件	有	有	有
结果比较方式	全文比较	全文比较	全文比较
题目类型	传统	传统	传统
是否打包	部分	否	否
运行内存上限	256 MB	512 MB	256 MB

二. 提交源程序文件名

对于C++语言	dishash.cpp	dissplay.cpp	dislct.cpp
对于C语言	dishash.c	dissplay.c	dislct.c
对于pascal语言	dishash.pas	dissplay.pas	dislct.pas

三. 编译命令 (不包含任何优化开关)

对于C++语言	g++ -o dishash dishash.cpp -lm -static	g++ -o dissplay dissplay.cpp -lm -static	g++ -o dislct dislct.cpp -lm -static
对于C语言	gcc -o dishash dishash.c -lm -static	gcc -o dissplay dissplay.c -lm -static	gcc -o dislct dislct.c -lm -static
对于pascal语言	fpc dishash.pas	fpc dissplay.pas	fpc dislct.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: 小机房老年机, 上述时限以此配置为准。(大雾)
- 4、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。
- 5、本卷并不保证题目难度有序, 请不要随意开题。

1. 小小迪的哈希树

(dishash.c/cpp/pas)

【题目背景】

小小迪是一个富有想象力的孩子。

【题目描述】

sxd 在给小小迪上 OI 课，正在讲树哈希，聪明的小小迪一看就会了。但是他觉得把“树”映射成“数”还不够，他决定把“数”映射向树。

现在，小小迪已经发明了这种神奇的算法，他已经能够把每一个数映射向一棵有 n 个节点的 m 叉树。

然而小小迪并没有采用什么方法来解决哈希冲突，于是他想问你最少需要多少个树才能保证这个算法出现哈希冲突。

要注意的是各个树之间形状不同，每个点之间没有区别，且 m 叉树的定义为每个节点度数 $\leq m$ （即儿子个数 $\leq m$ ）。

【输入格式】

一行两个正整数 n, m 表示询问 n 个节点的 m 叉树。

【输出格式】

一行一个整数 ans 答案，对 23333 取模。

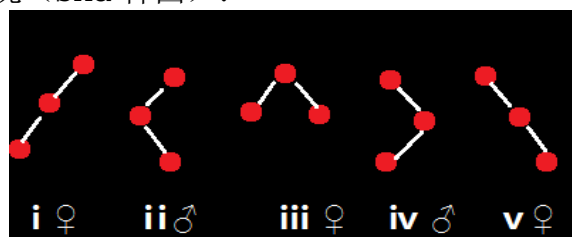
【输入输出样例1】

dishash.in	dishash.out
3 2	6

见选手目录下的 dishash/dishash1.in 和 dishash/dishash1.ans。

【输入输出样例1说明】

仅有以下五种情况（sxd 神图）：



由于是保证产生哈希冲突，所以说答案还要+1。

【输入输出样例2】

dishash.in	dishash.out
128 128	13714

见选手目录下的 dishash/dishash2.in 和 dishash/dishash2.ans。

【数据规模与约定】

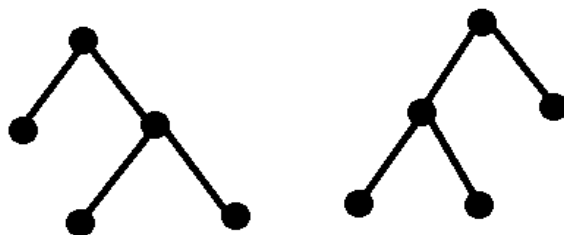
测试点	n	m	该子任务的分值
1 ~ 5	≤ 4	≤ 3	20
6 ~ 10	≤ 7	≤ 10	20
11 ~ 15	≤ 100	$= 2$	20
16 ~ 20	≤ 128	≤ 128	20
21 ~ 24	≤ 300	≤ 300	16
25 ~ 29	≤ 1000	≤ 1000	2
30 ~ 34	$\leq 10^9$	$\leq 10^9$	2

第25 ~ 29, 30 ~ 34个点打包测评, 其余每个点分值均为4。

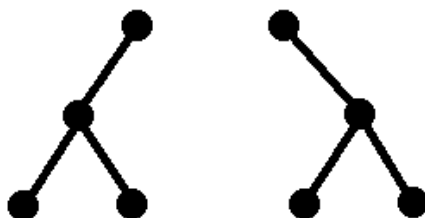
【解释与说明】

请务必仔细阅读**【题目描述】**中所给出的定义, n 个不同的节点构成二叉树下面解释一些常见的问题:

1. 下面两棵树的形态是不同的, 即不能交换子树。



2. 下面两棵子树的形态是不同的。



2. 小小迪的伸展树

(display.c/cpp/pas)

【题目背景】

小小迪是一个富有军事管理能力的孩子。

【题目描述】

众所周知，小小迪在基地里有一支庞大滑稽军，他们英勇善战，纪律严明，却也很爱吃滑稽果。

为了犒劳滑稽军，小小迪决定定期给这支滑稽军分滑稽果。

已知现在滑稽军有个 n 师，每个师的人数可能不相同，但小小迪并没有管这些，他给每个师的滑稽果数量是相等的。

为了保证各个师内部的公平正义，一个师内所有人的到的滑稽果必须一样多；由于滑稽果人人都爱吃，所以每个人会拿的尽量多。即如果发了 a 个滑稽果，该师中有 b 人，则每个人会分到 $\lfloor \frac{a}{b} \rfloor$ （如果 $a < b$ ，那么整个师就没人分到滑稽果了）。不难发现可能还有一些滑稽果多出来，这些滑稽果又会返还给小小迪给他吃。

小小迪想知道当他给每个师发给个滑稽果时一共有多少个滑稽果被返还给他。

当然，小小迪的军队也不是一成不变的，他的军队可能发生以下四种情况：

1. 新增了一个有 x 人的师。
2. 由于种种原因，有一个有 x 人的师解散了。
3. 有一个有 x 人的师实行裁军，将该师人数减少了 y 人。
4. 有一个有 x 人的师实行扩军，将该师人数增加了 y 人。

小小迪不能让一个师过于强大，因此他限制了每个师的人数上限 lim 。

当然，小小迪的滑稽果也并不是很多，他每次最多只可能给每个师 L 个滑稽果。

【输入格式】

第一行三个正整数 n, lim, L ，表示初始时刻不同人数军队的种数、每个师的人数限制和他每次最多只可能给每个师多少个滑稽果。

接下来行，每行包含两个整数 a_i 和 b_i ，表示初始时小小迪的军队中人数为 a_i 的师有 b_i 个。数据保证 a_i 各不相同。

接下来一个整数 m ，表示询问个数。

接下来 m 行，每行包含一个询问，格式如下：

询问	含义
add x	新增了一个有 x 人的师。
break x	一个有 x 人的师被解散了。
cut x y	一个有 x 人的师人数减少了 y 人。
expansion x y	一个有 x 人的师人数增加了 y 人。
query x	询问如果给每个师发 x 个滑稽果，小小迪一共收回多少个滑稽果。

【输出格式】

对于每个指令为query的行输出一行ans表示答案。

【输入输出样例1】

dissplay.in	dissplay.out
5 10 10	70
4 6	96
1 4	72
2 6	
7 6	
10 10	
10	
add 1	
cut 4 2	
cut 10 3	
query 3	
break 2	
expansion 1 4	
add 6	
cut 7 0	
query 5	
query 4	

见选手目录下的dissplay/dissplay1.in和dissplay/dissplay1.ans。

【输入输出样例2】

见选手目录下的dissplay/dissplay2.in和dissplay/dissplay2.ans。

【数据规模与约定】

测试点	n	m	b	lim	L
1 ~ 4	≤ 3000	≤ 3000	≤ 3000	≤ 5000	≤ 5000
5 ~ 7	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$	≤ 200
8 ~ 10	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$
11 ~ 14	$\leq 10^5$	$\leq 10^5$	$= 1$	$\leq 10^5$	$\leq 10^5$
15 ~ 18	5×10^4	5×10^4	5×10^4	5×10^4	5×10^4
19 ~ 20	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$

保证第8 ~ 10个点query的次数 ≤ 200 。

对于100%的数据保证生成方式随机，详见选手目录下

dissplay/make_data.cpp、dissplay/sj.cpp和dissplay/rand.h，以及编译好的dissplay/make_data.exe与dissplay/sj.exe。【输入输出样例2】即测试数据均由有该程序生成。

【解释与说明】

由于不开 long long 者而爆 0 者后果自负。

由于数据随机，请大胆猜想，争取大力**标程。

3. 小小迪的林克卡特树

(dislct.c/cpp/pas)

【题目背景】

小小迪是一个爱园艺的孩子。

【题目描述】

小小迪最喜欢的就是树了，可是他是一位新手，他现在只有一棵树。这是一棵以1为根节点的随机树。

这颗树不大（吗），只有 n 个节点。可是小小迪不喜欢一棵树，他想要的是一片森林。

所以小小迪会去掉这棵树上的一些边，使它成为森林，但是小小迪还需要把这些树种到他的花园中，他希望他花园里的树都是美观的。

小小迪对于美观的定义是：

一棵树的根节点到各个点的距离之和最小，如果存在有两个节点同时满足条件那么标号小的节点作为根节点。

小小迪要知道这些树的根节点来埋进土里，但他还想知道一些点现在的高度（即节点的深度，根节点的深度为0），因此会有两个操作：

1. 删除一条边。
2. 查询节点的高度。

小小迪会有 m 次操作，保证操作合法。小小迪当然不知道怎么做，但他手里有baoli的标程，所以他想请你把标程给那啥。

【输入格式】

第1行一个数 n, m 。

第2 ~ n 行两个数 x, y ，表示 x 和 y 之间有一条边（ $1 \leq x, y \leq n$ ）。

接下来 m 行每行表示一个操作。

一个操作有两个数 x, y （ $1 \leq x \leq 2, 1 \leq y \leq n$ ）：

- $x = 1$ ：删读入的第 y 条边。
- $x = 2$ ：查询的 y 高度。

保证输入合法，数据随机。

【输出格式】

对于所有 $x = 2$ 的操作，输出一行表示答案。

【输入输出样例1】

dislct.in	dislct.out
3 3	1
2 1	1
2 3	
2 3	
1 2	
2 2	

见选手目录下的dislct/dislct1.in和dislct/dislct1.ans。

【输入输出样例2】

dislct.in	dislct.out
7 7	1
2 1	1
3 1	0
1 4	0
1 5	1
1 6	
3 7	
2 6	
2 4	
1 6	
1 4	
2 7	
2 5	
2 3	

见选手目录下的dislct/dislct2.in和dislct/dislct2.ans。

【输入输出样例3】

见选手目录下的dislct/dislct3.in和dislct/dislct3.ans。

【数据规模与约定】

测试点	n, m
1 ~ 5	≤ 3000
6 ~ 8	≤ 6000
9 ~ 16	≤ 50000
17 ~ 20	≤ 300000

对于第9 ~ 10个点，保证 $x \neq 1$ 。

对于100%的数据，保证数据随机。