

全国青少年信息学奥林匹克竞赛

NOIP2023模拟

时间：8:30-12:00

题目名称	轰炸	修剪树木	树连接	字符串
题目类型	传统型	传统型	传统型	传统型
目录	bomb	pruning	tree	string
可执行文件名	bomb	pruning	tree	string
输入文件名	bomb.in	pruning.in	tree.in	string.in
输出文件名	bomb.out	pruning.out	tree.out	string.out
每个测试点时限	1.0秒	1.0秒	2.0秒	1.0秒
内存限制	512 MB	512MB	512MB	512MB
子任务数目	20	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于C++语言	bomb.cpp	pruning.cpp	tree.cpp	string.cpp
---------	----------	-------------	----------	------------

编译选项

对于C++语言	-lm -std=c++14 -O2
---------	--------------------

注意事项与提醒（请选手务必仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C++ 中主函数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。
- 提交的程序代码文件的放置位置请参照各省的具体要求。
- 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
- 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 程序可使用的栈内存空间限制与题目的内存限制一致。
- 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。
- 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。
- 终评测时所用的编译命令中不含编译选项之外的任何优化开关。

轰炸 (bomb)

【问题描述】

战狂也在玩《魔方王国》。他只会征兵而不会建城市，因此他决定对小奇的城市进行轰炸。

小奇有 n 座城市，城市之间建立了 m 条有向的地下通道。战狂会发起若干轮轰炸，每轮可以轰炸任意多个城市。

每座城市里都有战狂部署的间谍，在城市遭遇轰炸时，它们会通过地下通道撤离至其它城市。非常不幸的是，在地道里无法得知其它城市是否被轰炸，如果存在两个不同的城市 i, j ，它们在同一轮被轰炸，并且可以通过地道从城市 i 到达城市 j ，那么城市 i 的间谍可能因为撤离到城市 j 而被炸死。为了避免这一情况，战狂不会在同一轮轰炸城市 i 和城市 j 。注意：炸毁的城市还是能够到达的。

你需要求出战狂最少需要多少轮可以对每座城市都进行至少一次轰炸。

【输入格式】

第一行两个整数 n, m 。

接下来 m 行每行两个整数 a, b 表示一条从 a 连向 b 的单向边。

【输出格式】

输出一行仅一个整数表示答案。

【样例输入1】

```
5 4
1 2
2 3
3 1
4 5
```

【样例输出1】

```
3
```

【样例1解释】

一个最优方案为：第一轮轰炸城市 1, 4，第二轮轰炸城市 2, 5，第三轮轰炸城市 3。

如果轰炸轮数少于 3，那么 1, 2, 3 一定至少两个会被同时轰炸，那么就会使得间谍无法撤离。

【样例2】

见下发文件中。

【数据范围及约定】

对于 20% 的数据， $n, m \leq 10$ 。

对于 40% 的数据， $n, m \leq 1000$ 。

对于另外 30% 的数据，保证无环。

对于 100% 的数据， $n, m \leq 1000000$ 。

修剪树木 (pruning)

【问题描述】

你有一棵带边权的树（一个无环的连通图），但它太大了。您要通过删除一些边来修剪树，以便保留以下属性。对于剪枝后的树的每个连通分量 C ，存在一个节点 r_C ，使得对于 C 中的每个节点 v ， v 与 r_C 之间路径上的边的总权重至多为某个给定值 d 。

您碰巧喜欢浓密的树，因此您希望删除最小可能的总边权重以满足此属性。

【输入格式】

每个测试用例的第一行是 T ，表示接下来测试用例的数量。

每个测试用例的第一行包含两个整数 n 和 d ，表示树中有 n 个节点， d 与问题描述中指定的相同。

假设树中的节点按 0 到 $n - 1$ 之间的整数建立索引。

接下来是 $n - 1$ 行，每行包含三个整数 u 、 v 和 w 。这表示节点 u 和 v 之间有一个权重 w 的边。输入将使图无环且连通。没有一条边会被多次给出，并且输入中的任何边都不会是一个自环（即每个输入边的 $u \neq v$ ）。

【输出格式】

每个测试用例的输出只有一行包含一个整数，表示在删除一些边之后可以保留的边的最大总权重，以便剩余的图满足问题描述中描述的属性。

【样例输入1】

```
3
5 10
0 1 10
1 2 9
1 3 10
3 4 9
10 1
0 1 1
0 2 1
1 3 1
1 4 1
2 5 1
2 6 1
2 7 1
6 8 1
6 9 1
7 2
0 1 1
0 2 2
1 3 1
1 4 2
2 5 2
2 6 7
```

【样例输出1】

```
29
7
7
```

【样例1解释】

对于第一组数据，删除边 $(3, 4)$ ，选择 1 和 4 作为各自连通分量的 r_C 。

对于第二组数据，删除边 $(0, 2)(2, 6)$ ，选择 1, 2, 6 作为各自连通分量的 r_C 。

对于第三组数据，删除边 $(0, 1)(2, 6)$ ，选择 1, 2, 6 作为各自连通分量的 r_C 。

【样例2】

见下发文件中。

【数据范围及约定】

对于 100% 的测试点，保证 $T \leq 40, 1 \leq n \leq 100, 0 \leq d \leq 10^9$ ，并且，对于输入中的每个边 $u, v, w, 0 \leq u < v \leq n - 1$ 和 $1 \leq w \leq 10^7$ 。

树连接 (tree)

【问题描述】

大厨最近受小丑的影响，开始用他新买的一套道具东砍西砍。

这天下午，大厨决定从一棵 N 个节点的树上砍一些节点下来。大厨制定了这样的计划：

- 他会选择一对整数 (L, R) 满足 $1 \leq L \leq R \leq N$ ，然后砍掉编号不在 $[L, R]$ 中的所有节点；
- 大厨还会砍掉至少一个端点为上述被砍的节点的所有边。

对于整数对 (L, R) ，如果按照上述方案砍树之后，树仍然连通，那么我们称这个整数对合法。

大厨想知道对于他的这棵树而言，有多少合法整数对。不过大厨在后厨还有许多工作，因此，请你帮他计算。

【输入格式】

输入的第一行包含一个整数 T ，代表测试数据的组数。

接下来是 T 组数据。

每组数据的第一行包含一个整数 N 。

接下来 $N - 1$ 行，每行包含两个整数，代表树中的一条边。保证每组数据中给出的是一棵合法的树。

【输出格式】

对于每组数据，输出一行，包含一个整数，代表合法的数对数。

【样例输入1】

```
2
2
2 1
4
3 4
1 4
4 2
```

【样例输出1】

```
3
7
```

【样例1解释】

对于第一组数据， $(1, 1), (1, 2), (2, 2)$ 合法。

对于第二组数据， $(1, 1), (1, 4), (2, 2), (2, 4), (3, 3), (3, 4), (4, 4)$ 合法。

【样例2】

见下发文件中。

【数据范围及约定】

对于 30% 的测试点, $1 \leq N \leq 1000$, $T \leq 10$ 。

对于 100% 的测试点, 保证 $1 \leq T \leq 10^5$, $1 \leq N \leq 10^6$, 输入中每组数据的 N 之和 $\leq 10^6$ 。

字符串 (string)

【问题描述】

给出一个长度为 L 的字符串 s ，以及一个 n 个样本字符串的集合 S 。我们使用集合 S 通过以下方式消除字符串 s ：

任何时候当 S_i 在字符串 s 以连续子串出现时，可以删除（或不删除）。

每次删除后，将已删除子字符串的左侧和右侧连接起来，您将获得一个新的字符串 s 。

通过这种方式，尝试消除给定的字符串 s 以获得一个新的最小长度的字符串。你可以无限次删除。

【输入格式】

第一行包含字符串 s 。

第二行包含整数 n 。

在接下来 n 行中，第 i 行包含字符串 S_i 。

【输出格式】

输出一个整数，表示消除能得到的最小长度

【样例输入1】

```
aaabccd
3
abc
ac
aaa
```

【样例输出1】

```
2
```

【样例1解释】

$aaabccd$ 通过 abc 消除得到 $aacd$ 。

$aacd$ 通过 ac 消除得到 ad 。

【样例2】

见下发文件中。

【数据范围及约定】

对于 40% 的测试点，保证 $|L| < 9$ 。

对于 100% 的测试点，保证 $0 < |L| < 151, 0 < n \leq 30, 0 < |S_i| \leq 20$ 。

