文件名 light.cpp/in/out road.cpp/in/out cake.cpp/in/out abc.cpp/in/out

题目名称	星星灯（light）	星星路（road）	星星糕（cake）	星星树（abc）
题目类型	传统型	传统型	传统型	传统型
目录	light	road	cake	abc
可执行文件名	light	road	cake	abc
时间限制	1 秒	1 秒	1 秒	1 秒
内存限制	256MB	256MB	256MB	256MB
测试点数目	40	40	40	40
测试点是否等分	是	是	是	是

星星灯（light）

【题目描述】

红薯在解一个叫做“星星灯”的益智谜题！

在这个谜题中，你需要控制按行列排列的 $n \times m$ 的星星灯。每盏星星灯在最开始的时候有亮或灭两种状态，处于(i,j) 位置的星星灯的状态用 $a_{i,j}$ 表示，其中 0 表示灭，1 表示亮。按下某一盏灯的开关后，你可以将亮着的星星灯熄灭，灭着的星星灯亮起。

这个谜题允许红薯通过两种方式改变星星灯：

1.同时按下竖着的两个星星灯的开关，即位置为 (i,j) 和位置 为 $(i+1,j)$ 的灯的亮灭状态发生变化 ，$1 \le i \le n-1$；

2.同时按下横着的两个星星灯的开关，即位置为 (i,j) 和位置 为 $(i,j+1)$ 的灯的亮灭状态发生变化，$1 \le j \le m-1$。

红薯可以使用每种操作任意次，他想知道能否进行一系列操作，使得最后能让所有的星星灯亮起。如果能的话，输出 Yes，反之输出 No。

点我下载大样例：light.in，light.out，或者点我下载压缩包：light.zip

【输入格式】

第一行，包含一个整数 T，表示数据组数。

接下来给出 T 组样例，对于每一组样例：

第一行，包含两个正整数 n 和 m，表示谜题中有 n 行 m 列星星灯。

接下来的 n 行，每行包含一个长度为 m 的仅含 0,1 的字符串，表示星星灯初始的亮起状态。

【输出格式】

输出 T 行，每行输出一个 Yes 或 No 表示是否能让所有的星星灯亮起。

【样例 1 输入】

```
2
1 4
0011
2 4
0100
0111
```

【样例 1 输出】

```
Yes
Yes
```

【样例 2 输入】

```
1
2 4
0111
0011
```

【样例 2 输出】

```
No
```

【备注】

子任务	n	m	特殊性质	分值
1	=1	⩽ 2000	无	10
2	⩽ 2000	=1	无	10
3	⩽ 4	⩽ 4	T⩽ 100	20
4	⩽ 20	⩽ 20	T⩽ 3	20
5	⩽ 200	⩽ 200	T⩽ 3	10
6	⩽ 2000	⩽ 2000	无	30

对于全部数据，保证满足：$1 \leq \sum n, \sum m \leq 2000$，$1 \leq T$。

星星路（road）

【题目描述】

桔子决定出去旅游!

星星国有 n 个城市，$n-1$ 条道路，对于星星国的第 i 条道路，它由城市 x_i 指向城市 y_i ，表示可以从城市 x_i 走向城市 y_i。其中 1 号城市是星星国的首都，并且从星星国的任何一个城市出发，不能再沿着不重复的道路再走回这个城市。

因为桔子想尽可能感受星星国的风土人情，所以桔子选择从星星国的首都出发，沿着道路，到任何一个城市结束。

每个城市有一个标识字母，标识字母都是小写字母，记为 c_i，象征着这个城市的特色。

桔子从出发城市到结束城市经过的所有城市的标识字母会构成一条星星路（出发城市和结束城市也算），换句话说，星星路是一个由经过城市的标识字母构成的字符串。

桔子现在想知道，在她可以选择的所有星星路中，字典序最大的星星路是多少？

对于两个字符串 s_1, s_2，设长度分别为 l_1, l_2 $(l_1 \leq l_2)$ ，如果他们的前 $i-1$ 个字符都相同，在第 i 个字符有 $c_{1,i} < c_{2,i}$ ，则s_1的字典序小；若 $c_{1,i} > c_{2,i}$ ，则

s_2 的字典序小。

如果 s_2 的前 n 个字符和 s_1 相同，那么 s_1 的字典序比 s_2 小。

点我下载大样例：road.in，road.out，或者点我下载压缩包：road.zip

【输入格式】

第一行，包含一个正整数 n，表示城市的数量。

第二行，包含一个长度为 n 的仅包含小写字母的字符串，第 i 个字母表示城市 i 上的标识字母 c_i。

接下来 $n-1$ 行，每行包含两个整数 x_i, y_i 表示一条由城市 x_i 指向城市 y_i 的道路。

【输出格式】

输出一行，包含一个字符串，表示字典序最大的星星路。

【样例 1 输入】

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

【样例 1 输出】

```
zad
```

【样例 2 输入】

```
9
luaatinua
1 2
1 8
2 4
```

2 3
4 5
5 6
6 7
6 9

【样例 2 输出】

luatin

【备注】

子任务 1（20 分）：$n \leq 10$；

子任务 2（20 分）：$n \leq 2000$；

子任务 3（10 分）：$n \leq 10^6$，所有城市构成一条链；

子任务 4（10 分）：$n \leq 10^5$；　其实没有这一档

子任务 5（40 分）：$n \leq 10^6$。

对于全部的数据，保证有 $1 \leq n \leq 10^6$，标识字母均为小写字母。

整个图是弱连通的。

星星糕（cake）

【题目描述】

红薯开了一家专卖星星糕的蛋糕房!

红薯是一个刚入门的烘焙师，他得到了做星星糕的秘方:

用a 秒时间，用接水机接一碗水;

用b 秒时间，用磨面机磨一份面粉;

用c 秒时间，用揉面机将一碗水和一份面粉揉成一个面团（**如果没有水和面粉，则不能揉面)**;

用d 秒时间，用烤炉将一个面团烤为星星糕**（如果没有面团，则不能烤星星糕)**。

红薯可以同时持有多个水，面粉，面团，星星糕。

上述操作可以不连续。例如，假设红薯接水和磨面均需要 2 秒，那么他可以在时间区间 $[0,1)$ 和 $[2,3)$接水，时间区间$[1,2)$ 和 $[3,4)$磨面，并在第3 秒时得到一碗水，第 4 秒时得到一份面粉。

蛋糕房每 e 秒会来一位客人（第一个客人会在第 e 秒时到来），客人会等待 f 秒。如果红薯一直不让客人就餐，客人就会在 f 秒结束之后离开。

因为红薯的店刚开张，**所以红薯只有一个盘子，且最开始是干净的**。一份星星糕和一个干净盘子，可以让一个客人就餐。客人就餐需要 g 秒。客人**就餐结束后**，

你会得到 h 的收益，还会产生一个脏盘子。洗干净一个脏盘子需要 i 秒。

客人的行动与红薯的抉择无关。如果把红薯的行动看为时间轴，那么他在每一秒时可以选择一个会占用他时间的流程：接水，磨面，揉面团，烤星星糕，洗盘子。

今天红薯有 n 秒的经营时间，红薯想知道，他最多能赚多少钱呢？

注意：如果把时间看成数轴，那么红薯在第 1 秒做的事在 $[0,1)$ 进行，且第 1 秒来的客人在$[1,1]$来。

点我下载大样例：cake.in，cake.out，或者点我下载压缩包：cake.zip

【输入格式】

共一行，包含 10 个整数，分别为 n,a,b,c,d,e,f,g,h,i，其含义如题所述。

【输出格式】

共一行，包含一个整数，表示红薯能够得到的最大收益。

【样例 1 输入】

10 1 0 1 1 3 2 2 100 2

【样例 1 输出】

200

【样例 1 说明】

一共经营 10 秒，接水要 1 秒，磨面粉要 0 秒，揉面要 1 秒，烤星星糕要 1 秒，客人每 3 秒来一个，会等待 2 秒（第一个客人在第 5 秒仍然可以就餐），

就餐需要 2 秒，每个客人就餐让红薯得到 100 的收益，洗一个盘子需要 2 秒。

一种可能的最优策略是，红薯在时间 $[0,3)$ 烤出 1 个星星糕，在第 3 秒的时候为第 1 个客人上菜，然后用 2 秒接水磨面粉揉面。在第 5 秒的时候，红薯得到了一个脏盘子，得到了 100 的收益。他在第 6 秒烤好了星星糕，并迎来了第 2 位客人。红薯用 2 秒，在第 8 秒洗好了盘子，并为第 2 位客人上菜，此时恰是第 2 位客人等待的极限。在第 9 秒，红薯接水磨面粉，迎来了第 3 位客人。在第 10 秒，第 2 位客人产生 100 的收益，红薯揉了面团，营业结束，总收益为 200。

注意：我们不妨把所有事件画在一个时间轴上。那么第一秒来的客人相当于在 [1,1] 时间来。那么在数轴 $[0,3)$ 这段区间上红薯在做星星糕，在 $[3,3]$ 这个时间点客人来并开始用餐，在 $[5,5]$ 这个时间点客人吃完并离去.....在 $[9,10)$ 区间，红薯揉了面团，并在 [10,10] 这个时间点得到了客人的收益。

【样例 2 输入】

10 1 1 1 1 3 0 0 100 0

【样例 2 输出】

200

【样例 2 说明】

在本样例，红薯只能招待第 6 秒和第 9 秒的客人，因为他们完全不会等待。

【备注】

子任务	n	a,b,c,d,e,f,g,h,i	特殊性质	分值
1	≤10	$\leqslant 10^9$	无	20
2	≤2000	$\leqslant 10^9$	无	20
3	$\leqslant 10^5$	$\leqslant 10^9$	i=0	10
4	$\leqslant 10^5$	$\leqslant 10^9$	g=0	10
5	$\leqslant 10^5$	$\leqslant 10^9$	f=0	10
6	$\leqslant 10^5$	$\leqslant 10^9$	无	30

对于全部数据，保证满足：$1 \leq n \leq 10^5, 0 \leq a, b, c, d, f, g, h, i \leq 10^9, 1 \leq e \leq 10^9$。

星星树（abc）

【题目描述】

桔子有一棵星星树!

星星树是一张有 n 个节点，编号从 1 到 n，恰有 $n-1$ 条边的无向连通图。特别的，桔子认为 1 号是星星树的根。星星树的每个节点上都有一个标识字母 $\mathrm{sign_i}$，它是小写字母 a, b, c 的其中一个。

定义从 x 到 y 的路径，路径可以用其经过的点的编号来表示，即 $t_1(x), t_2, \ldots, t_k(y)$，那么一条路径要满足，该路径上的任意两个点的节点各不相同，并且 $\forall i \in [1, k-1]$，t_i 和 t_{i+1} 在图中有一条边相连。定义点 x 的深度为从 1 号点到点 x 的路径上点的个数，用 d_x 来表示。定义叶子为只有一条边的节点。

定义一个**无回溯**的“abc 链”是满足以下条件的有序三元组$< x, y, z >$，y在从x到z的路径上，$\mathrm{sign}_x = a, \mathrm{sign}_y = b, \mathrm{sign}_z = c$。我们发现**无回溯的**“abc 链”总可以在一条从 1 号点到叶子 x 的路径上取得。

桔子会用这棵星星树帮助她学英文，她想知道星星树上有多少个 **无回溯的**“abc 链”，但红薯指出这个问题实在是太简单了，他用了 $\frac{1}{20012143}$ 秒就解决了这个问题。桔子冥思苦想想到了一个 puls 版，于是她询问红薯**回溯不超过 k 次的**“abc 链”共有多少条。

一次回溯，允许在到达 x 后，重新选择 1 到 x 路径上的一点 y ，再从 y 选一个与它有边相连的、满足 $d_z > d_y$ 的点 z 走过去，直到走到叶子节点 λ。**此时，允许 “abc 链”在从 1 到 x 的路径上，和从 y 到 λ 的路径上共同取得，也即可以认为 "abc" 链出现在路径$1,\cdots,x,y,\cdots,\lambda$上。样例可能可以帮你更好的理解回溯的定义。**

红薯当然会解决这个问题，现在他要用这个问题来考考你了。

点我下载大样例：abc.in，abc.out，或者点我下载压缩包：abc.zip

【输入格式】

第一行，两个整数 n,k，表示节点个数和回溯次数。

第二行，一个长度为 n 的仅含小写字母 a,b,c 的字符串 S，S 的第 i 个小写字母表示 $sign_i$。

接下来 $n-1$ 行，每行两个整数 x,y 表示一条连接点 x 和 y 的双向边。

【输出格式】

输出一行一个整数，表示方案数。

【样例 1 输入】

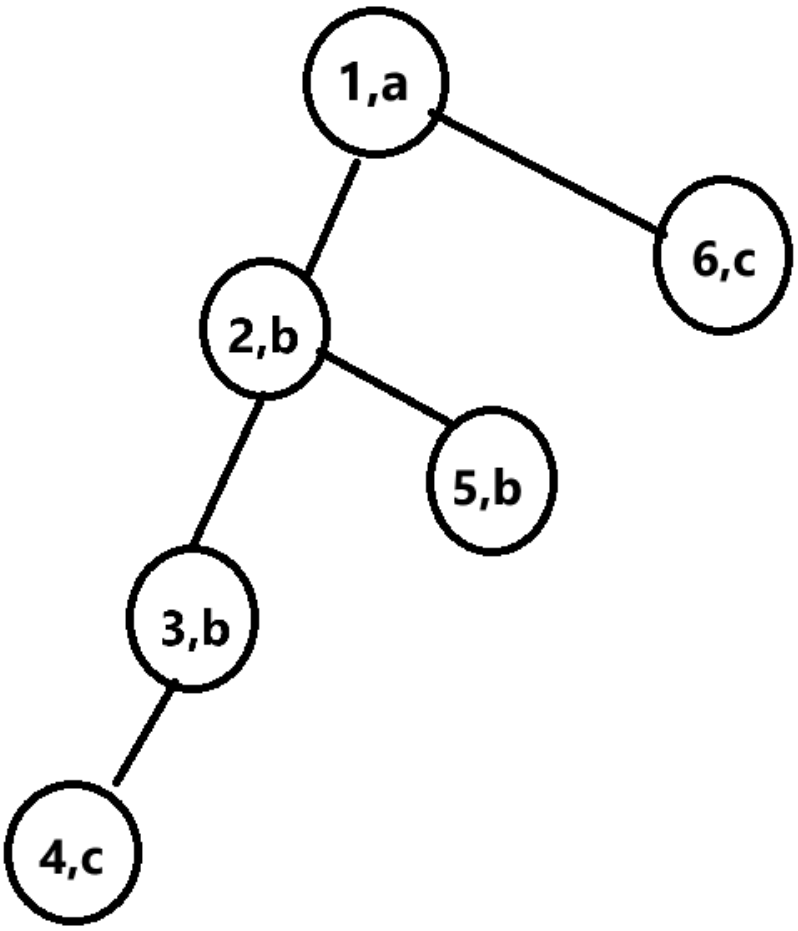
```
6 0
abbcbc
1 2
1 6
2 5
2 3
3 4
```

【样例 1 输出】

```
2
```

【样例 1 说明】

方案有 <1,2,4> 和 <1,3,4>



【样例 2 输入】

6 1
abbcbc
1 2
1 6
2 5
2 3
3 4

【样例 2 输出】

6

【样例 2 说明】

方案有 <1,2,4>, <1,3,4>, <1,5,4>; <1,2,6>, <1,3,6>, <1,5,6>。

对于 <1,5,4>, 可以走到 5 后回溯到 2 号点, 然后一直向 4 号点走, 直到取到 4。

对于 <1,2,6>, 可以拿了 1、2 号点上的字母后 后回溯到 1 号点 , 然后向 6 号点走。

【备注】

子任务 1（10 分）：$\mathrm{n} \leq 300$;

子任务 2（20 分）：$\mathrm{n} \leq 2000$;

子任务 3（5 分）：$\mathrm{n} \leq 10^5$，$\mathrm{k} = 0$;

子任务 4（5 分）：$\mathrm{n} \leq 10^5$，$\mathrm{k} = 1$;

子任务 5（5 分）：$\mathrm{n} \leq 10^5$，$\mathrm{k} = 2$;

子任务 6（5 分）：$\mathrm{n} \leq 10^5$，图是一条链，并且根的度数为 1;

子任务 7（5 分）：$\mathrm{n} \leq 10^5$，图是一张菊花图;

子任务 8（5 分）：$\mathrm{n} \leq 10^5$，保证树高不超过 $10 \times \log_2 n$;

子任务 9（10 分）：$\mathrm{n} \leq 10^4$;

子任务 10（30 分）：$\mathrm{n} \leq 10^5$;

对于全部的数据，保证 $1 \leq \mathrm{n} \leq 10^5, \mathrm{sign}_i \in \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}, \mathrm{k} \leq 2$。