

Задача АВ11. ЗАРАЗА

 5 сек.  1024 MB

Атина вече е пораснала и е една от VIP (Very Important Party-goer – много важен купонджия) личностите в страната Слория. Тъй като Слория има много ефективни проектанти, страната разполага с N града, свързани с $N - 1$ двупосочни пътища (U_j, V_j) помежду им, така че е възможно да се пътува между всеки два града, използвайки само тези пътища.

Въпреки това, миналия месец правителството в Слория установи, че се разпространява силно заразна инфекция и сега е загрижено за благосъстоянието на ВИП-овете. За тази цел правителството проследи точните планове на всички M купонджии: деня (от началото на Вселената), в който ще започнат да купонясват D_i , града S_i , от който ще започнат, и града T_i , в който ще приключат.

Тъй като модерният начин да се купонясва е всяка вечер да се прави парти в различен град, всеки купонджия i ще пристигне в Слория вечерта на ден D_i в града S_i и ще купонясва там през тази нощ. След това ще продължи, като се придвижва от град в град, изминавайки по един път на ден и следвайки най-краткия път до T_i , като всяка вечер присъства на парти в града, в който се намира в момента. След като купонясва в T_i , на следващата сутрин ще излети от Слория и това ще сложи край на неговото парти-пътешествие.

Освен това правителството на Слория е получило информация дали всеки купонджия е бил заразен в началото на своето парти-пътешествие, обозначено с I_i . Тъй като партитата са много социално занимание, ако някой е заразен на парти в даден град, всички останали, които купонясват в същия град в същата вечер, също се заразяват. Понеже засега няма лечение за инфекцията, всеки заразен остава заразен до края на своето парти-пътешествие.

Правителството на Слория иска да изчисли колко нощи всеки купонджия ще прекара заразен по време на своето парти-пътешествие. Ваша задача е да им помогнете, като реализирате ефективно решение на този проблем.

Детайли по имплементацията

Трябва да имплементирате функцията `solve`:

```
std::vector<int> solve(  
    std::vector<std::pair<int, int>> R,  
    std::vector<long long> D,  
    std::vector<int> S,  
    std::vector<int> T,  
    std::vector<bool> I  
);
```

- R : вектор от $N - 1$ двойки цели числа — двата града U_j и V_j , свързани с път j ;
- D : вектор от M цели числа — началният ден на купонджия i ;
- S : вектор от M цели числа — началният град на купонджия i ;
- T : вектор от M цели числа — крайният град на купонджия i ;
- I : вектор от M булеви стойности — началният статус на заразяване на купонджия i , където 1 означава “заразен” и 0 означава “незаразен”.

**XVII INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS
ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, БУРГАС 2026**

Тази функция ще бъде извикана веднъж за всеки тест и трябва да върне вектор от M цели числа — броя нощи, които купонджия i ще прекара заразен.

Ограничения

- $1 \leq N, M \leq 150000$
- $0 \leq D_i \leq 10^{12}$
- $0 \leq U_j, V_j, S_i, T_i < N$
- $I_i \in \{0, 1\}$

Подзадачи

Подзадача	Точки	N, M	Допълнителни ограничения
0	0	-	Примерен тест.
1	5	≤ 100	$D_i = 0$
2	6	≤ 5000	$D_i = 0$
3	7	≤ 5000	-
4	5	≤ 150000	$D_i = 10^6 \times i$
6	20	≤ 150000	$(U_j, V_j) = (j, j + 1)$ $D_i = 0$
7	19	≤ 150000	$(U_j, V_j) = (j, j + 1)$
8	18	≤ 100000	-
9	20	≤ 150000	-

Точките за дадена подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея.

**XVII INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS
 ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, БУРГАС 2026**

Примерен тест

Вход	Изход	Инфраструктура на Слопия	Обяснение на примера																																																																														
9 5 0 1 1 2 2 3 3 4 4 5 3 6 6 7 6 8 1 0 5 1 3 5 7 0 4 5 7 0 8 7 8 0 10 0 1 0	6 0 4 3 0		Парти-маршрутите на всички ВИП-ове са показани в следната таблица, като дните, в които всеки човек е бил заразен, са отбелязани с удебелен шрифт .																																																																														
			<table border="1"> <thead> <tr> <th>-</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> </tr> </thead> <tbody> <tr> <th>0</th> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <th>1</th> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <th>2</th> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <th>3</th> <td>2</td> <td>5</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <th>4</th> <td>3</td> <td>4</td> <td>5</td> <td>-</td> <td>-</td> </tr> <tr> <th>5</th> <td>4</td> <td>3</td> <td>4</td> <td>-</td> <td>-</td> </tr> <tr> <th>6</th> <td>5</td> <td>6</td> <td>3</td> <td>-</td> <td>-</td> </tr> <tr> <th>7</th> <td>-</td> <td>7</td> <td>6</td> <td>-</td> <td>-</td> </tr> <tr> <th>8</th> <td>-</td> <td>-</td> <td>7</td> <td>7</td> <td>-</td> </tr> <tr> <th>9</th> <td>-</td> <td>-</td> <td>-</td> <td>6</td> <td>-</td> </tr> <tr> <th>10</th> <td>-</td> <td>-</td> <td>-</td> <td>8</td> <td>0</td> </tr> <tr> <th>11</th> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> </tr> </tbody> </table>	-	0	1	2	3	4	0	-	-	-	-	-	1	0	-	-	-	-	2	1	-	-	-	-	3	2	5	-	-	-	4	3	4	5	-	-	5	4	3	4	-	-	6	5	6	3	-	-	7	-	7	6	-	-	8	-	-	7	7	-	9	-	-	-	6	-	10	-	-	-	8	0	11	-	-	-	-	1
-	0	1	2	3	4																																																																												
0	-	-	-	-	-																																																																												
1	0	-	-	-	-																																																																												
2	1	-	-	-	-																																																																												
3	2	5	-	-	-																																																																												
4	3	4	5	-	-																																																																												
5	4	3	4	-	-																																																																												
6	5	6	3	-	-																																																																												
7	-	7	6	-	-																																																																												
8	-	-	7	7	-																																																																												
9	-	-	-	6	-																																																																												
10	-	-	-	8	0																																																																												
11	-	-	-	-	1																																																																												

Формат на грейдъра

Формат на входа:

- $N M$
- $U_0 V_0$
- $U_1 V_1$
- ...
- $U_{N-2} V_{N-2}$
- $D_0 S_0 T_0 I_0$
- ...
- $D_{M-1} S_{M-1} T_{M-1} I_{M-1}$

Формат на изхода:

- Ans_0
- Ans_1
- ...
- Ans_{M-1}