



Задача Портокали



На картината: едит на капибара в горещ извор
от японска зоологическа градина с портокал на главата

В зоологическа градина някъде в Япония, пазачите решили да играят следната игра с капибарите:

Капибарите се намират в N горещи извора, номерирани с числата от 0 до $N-1$. Горещите извори са свързани с $N-1$ пътеки. Всяка пътека свързва два горещи извора, така че е възможно да стигнем от всеки горещ извор до всеки друг чрез пътеките. Казано по друг начин, капибарите се намират в структура, която е дърво (неориентиран свързан ацикличесен граф).

В началото има най-много една капибара във всеки горещ извор, но това може да се промени по време на играта.

Играта се състои от няколко (възможно е и безброй много) рундове. Всеки рунд има 2 етапа:

1. Пазачите ще хвърлят портокал в един от N -те горещи извора. Капибарите разбират точно в кой горещ извор е хвърлен портокалът.
2. Най-много една от капибарите може да се придвижи към съседен горещ извор. След това, ако няма капибара в горещия извор с портокала, то пазачите печелят, а капибарите губят. В противен случай портокалът се изяжда и играта продължава.

Начална конфигурация (множеството от горещи извори с капибари) е *безопасна*, ако пазачите не могат да спечелят игра с краен брой рундове, при условие че и пазачите, и капибарите играят оптимално.

За всяко K от 0 до N , намерете броя *безопасни* начални конфигурации, които имат точно K капибари. Понеже този брой може да е много голям, отпечатайте отговора по модул 998244353.

■ **Детайли по имплементацията** Трябва да имплементирате следната функция:

```
std::vector<int> solve(int N, std::vector<int> U, std::vector<int> V)
```

Функцията се извиква точно веднъж от грейдъра и трябва да върне `std::vector<int>` с размер $N+1$, съдържащ броя на безопасните начални конфигурации (по модул 998244353), така че да има точно K капибари за всяко $0 \leq K \leq N$.

Описание на параметрите на функцията:



- N - брой на горещите извори.
- U - `std::vector<int>` с размер $N - 1$, който съдържа едните краища на всяка от $N - 1$ пътеки.
- V - `std::vector<int>` с размер $N - 1$, който съдържа другите краища на всяка от $N - 1$ пътеки.

За всяко $0 \leq i < N - 1$, i -тата пътека свързва горещите извори U_i и V_i .

Не забравяйте да включите хедърния файл "oranges.h", иначе ще получите компилационна грешка!

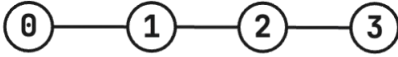
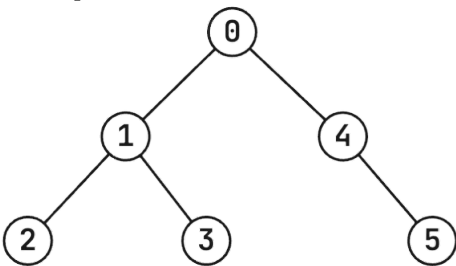
■ Ограничения

- ◆ $1 \leq N \leq 6000$
- ◆ За всяка пътека (U_i, V_i) , $0 \leq U_i, V_i < N, U_i \neq V_i$
- ◆ Гарантирано е, че дадените пътеки образуват дърво (свързан ацикличен граф).

#	Точки	Ограничения
1	4	Съществува горещ извор, който е директно свързан с всеки друг горещ извор.
2	11	Всеки горещ извор е директно свързан с най-много два други горещи извора.
3	14	$N \leq 10$
4	9	$N \leq 20$
5	33	$N \leq 200$
6	29	Няма допълнителни ограничения.



Примери

Вход	Изход	Обяснение
1	0 1	Единствената безопасна начална конфигурация е {0}.
4 0 1 1 2 2 3	0 0 4 4 1	Структурата, в която се намират капибарите, е следната:  Има 4 безопасни начални конфигурации с по две капибари: {0, 2}, {0, 3}, {1, 2}, {1, 3}. Всички начални конфигурации, които имат поне 3 капибари са безопасни.
6 0 1 1 2 1 3 0 4 4 5	0 0 0 0 11 6 1	Структурата, в която се намират капибарите, е следната:  Началната конфигурация {1, 3, 4} не е безопасна: През първия рунд пазачите ще хвърлят портокал в горещия извор 5. Капибарата в 4 трябва да се премести към 5. През втория рунд пазачите ще хвърлят портокал в горещия извор 2. Капибарата в 1 трябва да се премести към 2. През третия рунд пазачите ще хвърлят портокал в горещия извор 0. Понеже няма капибари, които могат да се придвижат директно до там, то пазачите печелят.
15 0 1 0 2 2 3 3 4 4 5 5 6 0 7 7 8 8 9 9 10 8 11 11 12 7 13 7 14	0 0 0 0 0 0 0 0 560 992 → 793 361 98 15 1	Структурата, в която се намират капибарите, е следната: 