Задача Twix

Header C++ `twix.h`

Във фабриката Left Twix карамелът се *излива* върху бисквитата. Във фабриката Right Twix карамелът се *спуска на каскади* върху бисквитата. Съперничеството между двете фабрики за производство на шоколадови десерти е в задънена улица още от 2012 г.

И двете фабрики опитали да добавят нови вкусове, но накрая предложили еднакъв асортимент. Вместо това те започнали да правят оферти: Left Twix въвела Bundle, с който получавате по едно копие от всеки изброен продукт; Right Twix отговорила с Choice, който позволява да изберете един продукт. И двете добавили Repeat, за да пестят място във формуляра за поръчки. Bundle и Choice могат да съдържат и други оферти, както и вкусове.

Описание на офертите

Всяка i-та оферта, която може да бъде закупена, е представена чрез ориентиран ацикличен граф с N_i върха. Всеки връх представлява оферта. Една оферта включва друга чрез препратка към нейния връх и може да включва само оферти с по-малък индекс. Последният връх на всеки граф представлява крайната оферта. Всяка оферта може да бъде използвана по различни начини в зависимост от изборите във върховете Choice, като така могат да се получат различни кошници с шоколадови десерти по различни начини.

- $\mathrm{Flavor}(c)$ има един възможен начин на използване, при който получавате един шоколадов десерт с вкус c.
- $\mathrm{Bundle}_K(X_0, \ldots, X_{K-1})$ ви дава по едно копие от всяка изброена оферта. Ако някоя оферта е изброена два пъти, получавате две нейни копия. При използването на такава оферта можете независимо да изберете как да използвате всяко копие, след което да поставите всички получени шоколадови десерти в една кошница.
- $\mathrm{Choice}_T(X_0, \ldots, X_{T-1})$ ви дава T възможности. При използването на такава оферта можете да изберете една възможност и да я използвате. Изборът на различни позиции се счита за различен начин на използване на офертата дори когато офертите на тези позиции са еднакви. (Вече е установено, че изборът между Left и Right е съвсем различен, дори ако те са „почти“ еднакви.)
- $\mathrm{Repeat}_R(X)$ е съкратен запис за R копия на една и съща оферта X измежду Bundle, Choice или Repeat. Всяко копие заема отделна позиция в съответния Bundle или Choice. Repeat може да се среща в Bundle, в Choice или в друг Repeat. Например,

$$\mathrm{Bundle}_2(\mathrm{Repeat}_3(X_0), X_1) = \mathrm{Bundle}_4(X_0, X_0, X_0, X_1).$$

  Същото правило важи и за Choice.

Последният връх, който представлява офертата за закупуване, не може да бъде Repeat.

Задача

Заради съперничеството между двете фабрики те взаимно се обвинили в *кражба*! Вие сте наети като инспектор по плагиатство, за да преброите двойките еквивалентни оферти, които могат да бъдат закупени.

Една кошница се определя от броя шоколадови десерти от всеки вкус, които съдържа. Например $(1, 1, 2)$ и $(2, 1, 1)$ описват една и съща кошница, която има 2 шоколадови десерта с вкус 1 и 1 шоколадов десерт с вкус 2.

Нека $w_D(b)$ е броят на различните начини за използване на офертата D, при които се получава кошницата b. Този брой е нула, ако D не може да произведе b независимо от направените избори. Две оферти D и E са **еквивалентни**, ако $w_D(b) = w_E(b)$ за всяка кошница b.

Дадени са M оферти, които могат да бъдат закупени. Пребройте двойките индекси (i, j), за които $0 \le i < j < M$ и офертите i и j са еквивалентни.

Интерфейс на функцията

Реализирайте следната функция, декларирана във файла `twix.h`.

```
long long count_equivalent(
    int M,
    std::vector<int> type,
    std::vector<int> value,
    std::vector<std::vector<int>> children,
    std::vector<int> root
);
```

Грейдерът извиква `count_equivalent` с M – броя на офертите, които могат да бъдат закупени, и четири масива. Вашето решение трябва да дефинира тази функция, без функция `main`.

Масивите `type`, `value` и `children` използват един и същ индекс на връх v. Индексите на върховете започват от 0 и продължават последователно през всички оферти. За $0 \le i \le M$ стойността `root[i]` е индексът на последния връх на i-тата оферта за закупуване.

Връх	`type[v]`	`value[v]`	`children[v]`
$\text{Flavor}(c)$	`FLAVOR = 0`	c	Празен списък.
$\text{Bundle}_K(X_0, \ldots, X_{K-1})$	`BUNDLE = 1`	0	Индексите на $X_0, \ldots, X_{K-1}$.
$\text{Choice}_T(X_0, \ldots, X_{T-1})$	`CHOICE = 2`	0	Индексите на $X_0, \ldots, X_{T-1}$.
$\text{Repeat}_R(X)$	`REPEAT = 3`	R	Индексът на X.

Списъците с деца съдържат индексите на върховете в първоначалния им вид, преди разгъването на Repeat. Един и същ индекс на връх може да се среща на няколко позиции. Във връх Repeat числото R е броят на копията на X.

Трябва да върнете броя на двойките еквивалентни оферти като стойност от тип `long long`.

Ограничения

- $1 \le M \le 10^5$, $N_i \ge 1$ и $\sum_i N_i \le 10^5$.
- За $\text{Flavor}(c)$ в i-тата оферта за закупуване е изпълнено $1 \le c \le N_i$. Една и съща стойност на c означава един и същ вкус във всички оферти.
- Bundle съдържа списък от $K \ge 1$ оферти, а Choice – списък от $T \ge 1$ оферти. Общата дължина на всички списъци в Bundle и Choice е най-много $2 \cdot 10^5$.
- За $\text{Repeat}_R(X)$ е изпълнено $1 \le R \le 10^9$.
- Всяка препратка сочи към връх с по-малък индекс в същия граф. Последният връх на оферта за закупуване не може да бъде Repeat.

#	Точки	Ограничения
1	16	$M \le 100$, $N_i \le 7$ и $K, T, R \le 3$.
2	10	Срещат се само върхове Flavor и Bundle.
3	14	Срещат се само върхове Flavor, Choice и Repeat.
4	30	Repeat не може непосредствено да съдържа друг Repeat.
5	30	Няма допълнителни ограничения.

Формат на примерния грейдер

Предоставеният файл `sample_grader.cpp` прочита входа в описания по-долу формат и отпечатва стойността, върната от `count_equivalent`.

Първият ред съдържа M. За всяка оферта за закупуване i следва ред с N_i, а след него – N_i реда, описващи последователно върховете ѝ.

Формат	Значение
`0 c`	$\text{Flavor}(c)$.
`1 K a0 ... a(K-1)`	Bundle_K със списък от офертите във върховете $a_0, \ldots, a_{K-1}$.
`2 T a0 ... a(T-1)`	Choice_T със списък от офертите във върховете $a_0, \ldots, a_{T-1}$.
`3 R a`	$\text{Repeat}_R(X)$, където X е офертата във връх a.

В примерните файлове индексите на върховете започват от 0 във всяка оферта.

Пример

Вход	Изход
3 5 0 1 0 2 2 2 0 1 3 2 2 1 1 3 7 0 1 0 2 1 2 0 0 1 2 0 1 1 2 1 1 3 2 3 2 3 2 4 5 6 0 1 0 2 1 2 0 0 1 2 0 1 1 2 1 1 2 3 2 3 4	1
3 3 0 1 2 2 0 0 1 2 1 1 3 0 1 1 2 0 0 2 2 1 1 4 0 1 1 2 0 0 3 4 1 2 1 2	1

Обяснение на пример 1

Нека $A = \text{Flavor}(1)$ и $B = \text{Flavor}(2)$. С (A, B) означаваме кошница с по един шоколадов десерт от всеки от двата вкуса.

1. **Оферта 0.** Bundle съдържа $\text{Repeat}_2(\text{Choice}_2(A, B))$, затова правите по един избор от всяко копие на Choice. Двойките избори са (A, A), (A, B), (B, A) и (B, B). Двете средни двойки дават една и съща кошница.
2. **Оферта 1.** След разширяването на Repeat, външният Choice става $\text{Choice}_4(\text{Bundle}_2(A, A), \text{Bundle}_2(B, B), \text{Bundle}_2($ и има четири опции:
   $\text{Bundle}_2(A, A)$, $\text{Bundle}_2(B, B)$, $\text{Bundle}_2(A, B)$ и $\text{Bundle}_2(A, B)$. Избирайки точно една от тези четири

позиции, ни дава един начин да получим (A, A), два начина да получим (A, B) и един начин да получим (B, B).

3. **Оферта 2.** Choice позволява да изберете позиция 1 за (A, A), позиция 2 за (A, B) или позиция 3 за (B, B). Всяка позиция дава съответната кошница по един начин.

Кошница	Начини в оферта 0	Начини в оферта 1	Начини в оферта 2
(A, A)	1	1	1
(A, B)	2	2	1
(B, B)	1	1	1

Това са всички възможни кошници. Оферти 0 и 1 имат еднакъв брой начини за получаване на всяка кошница. Оферта 2 се различава за (A, B), следователно $(0, 1)$ е единствената двойка еквивалентни оферти и отговорът е 1.