Zadanie: GRA
Gra w kolorowanie

XXX OI, etap II, dzień pierwszy. Plik źródłowy `gra.*` Dostępna pamięć: 256 MB. *15.02.2023*

Plansza do *gry w kolorowanie* składa się z n pól ponumerowanych liczbami od 1 do n. Niektóre z pól sąsiadują z innymi. Jest dokładnie $n-1$ par sąsiadujących pól i z każdego pola można dojść do dowolnego innego, przechodząc pomiędzy sąsiadującymi polami. (Można zatem powiedzieć, że plansza tworzy drzewo.)

W grze biorą udział dwaj gracze. Początkowo wszystkie pola planszy są białe, oprócz jednego pola pokolorowanego na czerwono (które należy do pierwszego gracza) i jednego innego pola pokolorowanego na niebiesko (które należy do drugiego gracza). Gracze wykonują swoje ruchy naprzemiennie. W swoim ruchu gracz wybiera jedno dowolne pole u pokolorowane na jego kolor, a następnie koloruje na ten kolor jedno dowolne *białe* pole v, które sąsiaduje z polem u. Grę przegrywa ten gracz, który nie będzie mógł wykonać ruchu.

Zastanawiamy się, dla jakich wyborów pól początkowych gracz pierwszy ma strategię wygrywającą, tzn. jest w stanie wygrać grę, niezależnie od ruchów gracza drugiego.

A konkretniej: dane są zbiory pól A i B. Napisz program, który obliczy, ile jest takich par różnych pól (a, b), że początkowe czerwone pole o numerze a należy do zbioru A, początkowe niebieskie pole o numerze b należy do zbioru B, a gracz pierwszy ma strategię wygrywającą.

Program powinien sobie też radzić z aktualizacjami zbiorów A i B. Każda z q aktualizacji dodaje bądź usuwa jedno pole z jednego ze zbiorów A lub B. Należy wypisać liczbę szukanych par (a, b) po każdej aktualizacji.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita n ($1 \leq n \leq 500\,000$) – liczba pól na planszy.

W kolejnych $n-1$ wierszach znajduje się opis planszy; i-ty z tych wierszy zawiera dwie liczby całkowite u i v ($1 \leq u, v \leq n$), które oznaczają, że pola o numerach u oraz v sąsiadują ze sobą.

W kolejnym wierszu znajdują się trzy liczby S_A, S_B i q ($1 \leq S_A, S_B \leq n$, $0 \leq q \leq 500\,000$) oznaczające kolejno: początkowy rozmiar zbioru A, początkowy rozmiar zbioru B oraz liczbę aktualizacji.

Następny wiersz zawiera ciąg S_A różnych liczb ze zbioru $\{1, \ldots, n\}$, oznaczających numery pól należących do zbioru A. Jeszcze kolejny wiersz zawiera ciąg S_B różnych liczb ze zbioru $\{1, \ldots, n\}$, oznaczających numery pól należących do zbioru B.

W kolejnych q wierszach znajduje się opis aktualizacji zbiorów; i-ty z tych wierszy składa się kolejno z dwóch znaków z, t oraz liczby całkowitej w, rozdzielonych pojedynczymi odstępami ($z \in \{\texttt{A}, \texttt{B}\}$, $t \in \{\texttt{+}, \texttt{-}\}$, $1 \leq w \leq n$). Znak z oznacza zbiór, na którym operujemy; znak t oznacza typ wykonywanej operacji (znak `+` oznacza dodanie pola do zbioru, a znak `-` oznacza usunięcie pola ze zbioru); natomiast liczba w oznacza numer wierzchołka, który jest dodawany lub usuwany. Możesz założyć, że każda aktualizacja zmienia zbiór (czyli tuż przed wykonaniem operacji `+` pola w nie było w zbiorze, a tuż przed operacją `-` pole w było w zbiorze).

Zarówno na początku, jak i w trakcie procesu aktualizacji zbiory A i B *nie muszą* być rozłączne. Przypomnijmy, że w zliczanych parach pól (a, b), takich że $a \in A$ i $b \in B$, musi zachodzić $a \neq b$.

Wyjście

Twój program powinien wypisać na wyjście dokładnie $q+1$ wierszy; i-ty z nich powinien zawierać jedną liczbę całkowitą, oznaczającą liczbę szukanych par (a, b) dla zbiorów A i B po pierwszych $i-1$ aktualizacjach. (W szczególności pierwszy wiersz powinien zawierać liczbę par dla początkowych zbiorów A i B.)

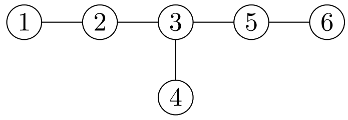
Przykład

Dla danych wejściowych:

```
6
1 2
2 3
3 4
3 5
5 6
1 2 1
1
5 6
A + 2
```

poprawnym wynikiem jest:

```
1
3
```



Wyjaśnienie przykładu: Początkowo $A = \{1\}$ i $B = \{5, 6\}$. Mamy dwie możliwości wyboru początkowych pól (a, b): parę $(1, 5)$ i parę $(1, 6)$. Dla pary $(1, 5)$ gracz pierwszy musi pokolorować pole 2, gracz drugi koloruje pole 3, i gracz pierwszy nie może wykonać ruchu. Dla pary $(1, 6)$ gracz pierwszy koloruje pole 2, gracz drugi musi pokolorować pole 5, gracz pierwszy koloruje pole 3, i gracz drugi nie może wykonać ruchu. Zatem gracz pierwszy ma strategię wygrywającą jedynie dla pary $(1, 5)$; zatem odpowiedzią jest 1.

Po uaktualnieniu zbiorów mamy $A = \{1, 2\}$ i $B = \{5, 6\}$. Gracz pierwszy ma strategię wygrywającą dla par $(1, 5)$, $(2, 5)$ i $(2, 6)$; zatem odpowiedzią jest 3.

Testy „ocen”:

- `1ocen`: $n = 10$, $q = 0$, pole $i > 1$ jest połączone z polem 1; $A = \{1, 2, 3\}$, $B = \{4, 5, 6\}$; wynik to 9.
- `2ocen`: $n = 200$, $q = 0$, pole $i > 1$ jest połączone z polem $i - 1$; $A = \{1, 2, 3\}$, $B = \{1, 2, 3\}$; wynik to 3.
- `3ocen`: $n = 2000$, $q = 0$, pole $i > 1$ jest połączone z polem $\lfloor \frac{i}{2} \rfloor$; $A = B = \{1, 2, \ldots, n\}$; wynik to 2 411 948.
- `4ocen`: $n = 500\,000$, $q = 0$, pole $i > 1$ jest połączone z polem $\lfloor \frac{i}{2} \rfloor$; $A = B = \{1, 2, \ldots, n\}$; wynik to 150 744 198 828.
- `5ocen`: $n = 500\,000$, $q = 1$, pole $i > 1$ jest połączone z polem $i - 1$; $A = \{1, 2, 3\}$, $B = \{1\}$; zapytanie usuwa pole 1 ze zbioru B.
- `6ocen`: $n = 500\,000$, $q = 1$, pole $i > 1$ jest połączone z polem $i - 1$; $A = \{1, 2, 3\}$, $B = \{1, 2, 3\}$; zapytanie usuwa pole 3 ze zbioru B.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Ponadto w każdym podzadaniu istnieją grupy testów warte łącznie co najmniej połowę punktów, w których n jest nieparzyste.

Podzadanie	Dodatkowe ograniczenia	Liczba punktów
1	$q = 0$, $n \leq 10$	8
2	$q = 0$, $n \leq 200$	10
3	$q = 0$, $n \leq 2000$	18
4	$q = 0$	30
5	zawsze $z =$ `B` (zbiór A się nie zmienia)	16
6	brak dodatkowych ograniczeń	18