

Misja: Potwór (misja-potwor)

Limit pamięci: 512 MB

Limit czasu: 4.00 s

– Gdybym tylko wzmocnił fale swego mózgu z supergenialnych do supermegagenialnych! – zachwycał się Kowalski.

- Ja myślałem, że twoje fale mózgowe są już pod niebo! – odpowiedział Szeregowy.
- I myślałeś, że w lodówce mieszka ludzik, co włącza i gasi światło.
- Kiedyś go dopadnę!

Marcel wie, że w lodówce nie mieszka, żaden ludzik co zapala i gasi światło, no ale Marcel nie jest pingwinem, tylko człowiekiem co jeździ swoim Nissanem 370Z. Na dodatek Marcel jest Informatykiem, który w antycznych czasach buraków i górtek, zdobył finał OI. O tym, że nasz bohater jest całkiem biegły w te klocki może zaświadczyć jego sposób implementacji algorytmu Union-Find:

```
int p[SIZE];
int counter = 0;

int Find(int u){
    counter++;
    return p[u] == u ? p[u] : p[u] = Find(p[u]);
}

void Union(int x, int y){
    p[Find(x)] = Find(y);
}

void Init(int n){
    for (int i = 0; i <= n; i++)
        p[i] = i;
}
```

No ale może szkoda rzucać bezwartościowe słowa na wiatr. Czas zająć się właściwą treścią zadania. Marcel zawsze bał się potworków. Oczywiście nie takich co mają futerko, groźne kły i ostre jak brzytwa pazury. Marcel boi się straszliwych problemów informatyczno-algorytmicznych. Przykładowo przez wiele lat swojego życia próbował dowiedzieć się, dlaczego powyższa implementacja algorytmu Union-Find działa w $\mathcal{O}(N \log N)$. Niestety Marcel nie jest burakiem i ten problem zwyczajnie go przerósł. Zagadka ta była jednak na tyle nurtująca, że postanowił zrobić absolutnie wszystko, by tylko poznać odpowiedź na nią. Pomyślał sobie, kto może znać rozwiązanie nurtującego go problemu i przypomniał sobie o największym geniuszu, jaki stąpał po tej Ziemi – Kowalskim. Już po chwili, za swoje ciężko zarobione pieniądze z organizacji konkursów informatycznych, kupił bilet do Stanów Zjednoczonych. W Central Park Zoo spotkało go jednak rozczarowanie. Pingwiny wcale nie były zbyt rozgarnięte i nawet pomimo dokładnych tłumaczeń zdawały się wcale nie rozumieć problemu Marcela.

Marcel porzucił już wszelkie nadzieje. Uwierzył, że powyższa implementacja algorytmu Union-Find działa w $\mathcal{O}(1)$ i nic nie jest w stanie go przekonać, że jest inaczej (warto tutaj zaznaczyć, że Marcel nie wierzy w jakieś odwrotne funkcje Ackermanna). Chyba że ktoś poda mu twardy dowód na to, że jednak istnieje przypadek, w którym powyższa implementacja algorytmu Union-Find wykonuje $\mathcal{O}(N \log N)$ operacji.

Twoim zadaniem będzie więc napisanie programu, który ułoży test zdolny przekonać Marcela do tego, że jego implementacja działa jednak w czasie $\mathcal{O}(N \log N)$.

Wejście

W pierwszym i jedynym wierszu wejścia znajduje się jedna liczba naturalna M , oznaczająca liczbę wywołań funkcji Find, którą należy osiągnąć.

Wyjście

W pierwszym wierszu wyjścia powinna się znaleźć jedna liczba naturalna Q oznaczająca liczbę operacji do wykonania, w kolejnych Q wierszach powinny się znaleźć opisy operacji w postaci słowa `Union`, pojedynczego odstępu i dwóch liczb X_i, Y_i , oddzielonych pojedynczym odstępem i odpowiadających argumentom wywołania funkcji `Union` lub w postaci słowa `Find`, pojedynczego odstępu i liczby X_i odpowiadającej argumentowi wywołania funkcji `Find`.

Ograniczenia

$1 \leq Q, X_i, Y_i \leq 1\,000\,000$.

Ocenianie

W tym zadaniu są tylko dwa testy, przykładowy i główny. W teście przykładowym $M = 10$, można za ten test otrzymać 0 punktów. W drugim właściwym teście $M = 10\,000\,000$, można za niego otrzymać wszystkie punkty za to zadanie.

Przykład

Wejście

10

Wyjście

```
7
Union 1 2
Union 2 3
Union 3 4
Find 1
Union 6 5
Union 5 4
Find 6
```

Wyjaśnienie

Po wykonaniu wszystkich operacji zmienna `counter`, odpowiadająca liczbie wywołań funkcji `Find`, będzie ustawiona na 17; wystarczy to do zaliczenia testu przykładowego.

Autor zadania uspokaja i na podstawie wykonanych testów twierdzi, że powyższa implementacja jest praktycznie nierozróżnialna czasowo od tej, która dodatkowo używa optymalizacji mniejszy do większego. Przynajmniej na zawodach programistycznych.

Niezbędny przypis: postacie z treści pochodzą z serialu "Pingwiny z Madagaskaru".