

Dostępna pamięć: 256MB

Usuwanie krawędzi

Dane jest drzewo, z którego po kolei usuwamy krawędzie. Proszę podać, po każdym usunięciu, rozmiary nowo powstałych spójnych składowych. Rozmiarem spójnej składowej nazywamy liczbę jej wierzchołków.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($2 \leq n \leq 200\,000$) oznaczająca liczbę wierzchołków drzewa. Wierzchołki drzewa numerujemy liczbami całkowitymi od 1 do n . W $n - 1$ kolejnych wierszach znajdują się opisy krawędzi – po dwie liczby całkowite u i v oznaczające istnienie krawędzi między wierzchołkami o nr u i o nr v . Krawędzie numerujemy liczbami całkowitymi od 1 do $n - 1$, zgodnie z kolejnością na wejściu.

W kolejnym wierszu zakodowana jest kolejność usuwania krawędzi – $n - 1$ liczb całkowitych a_1, a_2, \dots, a_n które oznaczają, że i -tą usuwaną krawędzią jest krawędź o numerze

$$a_i \oplus b_1 \oplus c_1 \oplus b_2 \oplus c_2 \oplus \dots \oplus b_{i-1} \oplus c_{i-1},$$

gdzie (b_j, c_j) są rozmiarami spójnych składowych powstałych po usunięciu j -tej krawędzi. Operacja \oplus to xor bitowy.

Taki format wejścia, w moim zamiarze, ma wymusić wyliczanie odpowiedzi *on-line*, tzn. i -te zapytanie dostępne jest dopiero po udzieleniu odpowiedzi na poprzednich $i - 1$ zapytań.

Wyjście

Na wyjście proszę wypisać $n - 1$ par liczb całkowitych – rozmiary spójnych składowych powstałych po usunięciu i -tej krawędzi. Proszę najpierw wypisać rozmiar nie większej składowej a później nie mniejszej.

Przykład

Wejście	Wyjście
7	2 5
1 2	2 3
1 3	1 2
3 4	1 1
2 5	1 1
2 6	1 1
1 7	
2 6 3 3 6 1	