

Zadanie: CEN

Centroid

polish

ONTAK, dzień 6. Dostępna pamięć: 256 MB. Limit czasu: 20 s.

3.07.2025

Dane jest pewne nieznanne drzewo o **nieparzystej** liczbie wierzchołków. Chcemy znaleźć jego centroid*. Aby to zrobić, mamy do dyspozycji zapytania – każde zapytanie bierze 3 wierzchołki i zwraca, czy jeden z nich leży na najkrótszej ścieżce między pozostałymi dwoma. Jeśli tak, podaje, który to wierzchołek.

W każdym teście, należy rozwiązać t różnych przypadków testowych. Łącznie we wszystkich przypadkach razem możesz użyć co najwyżej q zapytań.

Interakcja

To zadanie jest interaktywne. Należy napisać program, który będzie znajdował centroidy, używając do tego dostarczonej biblioteki. Aby użyć biblioteki, należy wpisać w swoim programie:

```
#include "cenlib.h"
```

Twój program zostanie skompilowany/uruchomiony za pomocą komendy:

```
g++ -O3 -static cen.cpp cenlib.cpp -std=c++20
```

Do dyspozycji masz następujące funkcje:

- `int give_t()` – podaje liczbę przypadków testowych t .
- `int give_n()` – daje w wyniku liczbę wierzchołków drzewa n . Jest to liczba nieparzysta, jej wartość zależy od subtasku.
- `int give_q()` – daje w wyniku maksymalną liczbę zapytań q . Jeśli Twój program zada więcej zapytań, otrzyma werdykt *Błędna odpowiedź*.
- `int give_subtask()` – daje w wyniku numer podzadania (patrz sekcja *Ocenianie*). W przypadku testów przykładowych jej wynikiem jest 0.
- `int query(int a, int b, int c)` – tej funkcji możesz użyć, aby dowiedzieć się, czy a , b i c leżą na jednej ścieżce i jeśli tak, który z nich pomiędzy pozostałymi. Wynikiem funkcji jest numer środkowego wierzchołka lub 0 gdy wierzchołki nie leżą na jednej ścieżce. Jeśli Twój program wywoła tę funkcję z którymś z parametrów spoza zakresu $[1, n]$, otrzyma werdykt *Błędna odpowiedź*.
- `void answer(int v)` – podaje centroid v jako odpowiedź. Wywołanie jej kończy dany przypadek testowy. Wszystkie następne wywołania funkcji `query` będą dotyczyły nowego drzewa. Kiedy zostanie wywołana t razy, kończy działanie programu.

Twój program nie może otwierać żadnych plików ani używać standardowego wejścia i wyjścia. Może on korzystać ze standardowego wyjścia diagnostycznego (`stderr`), jednak pamiętaj, że zużywa to cenny czas.

Przykładowa interakcja:

Wywołana funkcja	Wynik	Opis
<code>give_t()</code>	1	$t = 1$, zatem jest tylko jeden przypadek testowy
<code>give_q()</code>	100	limit zapytań $q = 100$
<code>give_subtask()</code>	0	jeden z testów przykładowych
<code>give_n()</code>	5	$n = 5$, ukryte drzewo ma krawędzie $(1, 2)$, $(2, 3)$, $(2, 4)$, $(4, 5)$
<code>query(1, 2, 3)</code>	2	
<code>query(2, 3, 4)</code>	2	
<code>query(1, 3, 4)</code>	0	wierzchołki 1, 3, 4 nie leżą na jednej ścieżce
<code>query(2, 4, 5)</code>	4	
<code>query(1, 2, 4)</code>	2	
<code>query(1, 4, 5)</code>	4	
<code>answer(2)</code>	—	da się odzyskać drzewo z zadanych zapytań

Testy przykładowe:

0a: $n = 3$, $t = 1$

0b: $n = 3$, $t = 2$

0c: $n = 5$, $t = 1$

*Centroid to taki wierzchołek, że po usunięciu go z drzewa, każda spójna składowa ma rozmiar nie większy niż połowa rozmiaru drzewa.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Podzadania różnią się limitami czasu.

Podzadanie	Ograniczenia	Limit czasu	Punkty
1	$n = 3, t = 100, q = 100$	0.5 s	2
2	$n = 999, t = 10, q = 10\,000\,000$	10 s	8
3	$n = 999, t = 100, q = 10\,000\,000$, drzewo jest ścieżką	10 s	12
4	$n = 999, t = 100, q = 10\,000\,000$	10 s	17
5	$n = 49\,999, t = 50, q = 15\,000\,000$, drzewo jest ścieżką	20 s	18
6	$n = 9\,999, t = 50, q = 20\,000\,000$	20 s	12
7	$n = 29\,999, t = 25, q = 20\,000\,000$	20 s	19
8	$n = 29\,999, t = 50, q = 22\,000\,000$	20 s	12

Wszystkie drzewa są ustalane przez bibliotekę na samym początku działania programu, przed odpowiadaniem na zapytania.

Eksperymenty

W sekcji *Pliki i testy* w SIO możesz znaleźć przykładową implementację biblioteki oraz testy przykładowe w formacie przyjmowanym przez tę bibliotekę.

Przykładowa biblioteka w pierwszym wierszu wejścia wczytuje kolejno t , q , s , gdzie s jest numerem podzadania. Następnie powinno znaleźć się t opisów drzew. Każdy opis powinien zawierać liczbę n oraz listę krawędzi.

Implementacja biblioteki używanej do oceny Twoich rozwiązań może się różnić od przykładowej.