

Zadanie: SIE

Sieć



Eliminacje do IOI, dzień pierwszy. Plik źródłowy sie.* Dostępna pamięć: 512 MB. 11.08.2020

Bajtazar jest administratorem w bajtockiej firmie BajtKom. W firmie znajduje się n serwerów, a sieć połączeń między nimi ma topologię drzewa, tzn. z każdego serwera można przesłać komunikat do każdego innego serwera (bezpośrednio lub pośrednio) za pomocą dokładnie jednej ścieżki połączeń (bez zawracania).

Ostatnio istotnym wyzwaniem dla Bajtazara są wirusy komputerowe, które pojawiają się w sieci firmowej na skutek ataków hakerów z wrogiego państwa Bitocji. Każdy wirus ma określony *zasięg* rażenia d . Gdy wirus o zasięgu d zaatakuje dany serwer, to zakłóca pracę tego serwera oraz wszystkich innych serwerów, do których da się z niego dotrzeć, korzystając z co najwyżej d bezpośrednich połączeń.

Bajtazar ma do dyspozycji program antywirusowy. Uruchomienie tego programu na danym serwerze wyrzuca z niego wszystkie wirusy, które go zainfekowały. W szczególności powoduje to, że wirusy te przestają zakłócać pracę tych serwerów, których pracę zakłócały wcześniej. Niestety, program antywirusowy Bajtazara nie zabezpiecza serwerów na przyszłość: serwer, który został odwirusowany przez Bajtazara, w przypadku ataku kolejnego wirusa zostaje ponownie zainfekowany.

Twoim zadaniem jest napisanie biblioteki, która pomoże zbierać dane o bezpieczeństwie w sieci BajtKomu. Będzie ona otrzymywać informacje o atakach wirusów oraz o uruchamianiu programów antywirusowych i ma za zadanie informować o *ryzyku* w przypadku przesyłania komunikatu między zadanymi parami serwerów. Ryzyko to określone jest jako liczba serwerów na trasie komunikatu (wliczając serwer początkowy i docelowy), których praca zakłócana jest przez wirusy.

Na koniec dodajmy, że kable łączące serwery w firmie są starej daty i łatwo tracą siłę sygnału. Z tego też względu komunikacja jest ograniczona do ścieżek o długości co najwyżej 500 (jest to jednocześnie maksymalny zasięg rażenia wirusa, jak i maksymalna odległość przesłania komunikatu).

Komunikacja

W tym zadaniu powinieneś napisać bibliotekę w języku C++, która będzie komunikowała się z programem oceniającym. Twoja biblioteka musi udostępniać (implementować) następujące funkcje, które będą wywoływane przez program oceniający (jeśli chcesz, Twoje rozwiązanie może zawierać także inne funkcje):

- **inicjuj(n)**
Ta funkcja zostanie wywołana tylko raz, na początku działania programu. Jej wywołanie informuje bibliotekę o liczbie serwerów n . Serwery są ponumerowane od 1 do n .
 - C++: `void inicjuj(int n);`
- **polaczenie(a, b)**
Ta funkcja zostanie wywołana $n - 1$ razy bezpośrednio po wywołaniu funkcji `inicjuj`. Jej wywołanie informuje bibliotekę o istnieniu bezpośredniego połączenia między serwerami o numerach a i b . O istnieniu każdego bezpośredniego połączenia zostaniesz poinformowany dokładnie raz.
 - C++: `void polaczenie(int a, int b);`
- **wirus(a, d)**
Wywołanie tej funkcji informuje bibliotekę o tym, że wirus zaraża serwer o numerze a ($1 \leq a \leq n$) i zaczyna zakłócać pracę wszystkich serwerów, które są w odległości co najwyżej d ($0 \leq d \leq 500$) bezpośrednich połączeń od tego serwera.
 - C++: `void wirus(int a, int d);`
- **antywirus(a)**
Wywołanie tej funkcji informuje bibliotekę o tym, że program antywirusowy usuwa wszystkie wirusy z serwera o numerze a ($1 \leq a \leq n$).
 - C++: `void antywirus(int a);`
- **ryzyko(a, b)**
Ta funkcja jest zapytaniem biblioteki o ryzyko przesłania komunikatu. Jej wywołanie powinno dać w wyniku liczbę serwerów na ścieżce łączącej serwery o numerach a i b ($1 \leq a, b \leq n, a \neq b$), których praca zakłócana jest przez wirusy. Możesz założyć, że ścieżka ta składa się z co najwyżej 500 bezpośrednich połączeń.

```
- C++: int ryzyko(int a, int b);
```

Twój program **nie może** czytać żadnych danych (ani ze standardowego wejścia, ani z plików). **Nie może** również nic wypisywać do plików ani na standardowe wyjście. Może pisać na standardowe wyjście diagnostyczne (`stderr`) – pamiętaj jednak, że zużywa to cenny czas. **Nie deklaruj** też funkcji `main`.

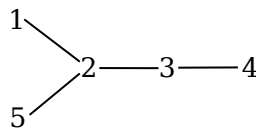
Kompilacja i uruchomienie

Twoja biblioteka `sie.cpp` zostanie skompilowana z programem oceniającym za pomocą polecenia

```
g++ -O3 -static sie.cpp siegrader.cpp -std=c++17
```

Przykładowy przebieg programu

Poniższa tabela przedstawia przykładowy ciąg wywołań funkcji wraz z poprawnymi wynikami wywołań funkcji `ryzyko`. Struktura sieci połączeń odpowiadająca temu przykładowi jest przedstawiona na rysunku.



Wywołanie funkcji	Wynik	Wyjaśnienie
<code>inicjuj(5)</code>	-	$n = 5$
<code>polaczenie(1, 2)</code>	-	połączenie między serwerami 1 i 2
<code>polaczenie(2, 3)</code>	-	połączenie między serwerami 2 i 3
<code>polaczenie(3, 4)</code>	-	połączenie między serwerami 3 i 4
<code>polaczenie(2, 5)</code>	-	połączenie między serwerami 2 i 5
<code>wirus(5, 1)</code>	-	wirus zaraża serwer 5, zakłócając pracę tego serwera oraz serwera 2, który jest w odległości 1 od niego
<code>ryzyko(1, 3)</code>	1	na ścieżce łączącej serwery 1 i 3 zakłócona jest praca serwera 2
<code>wirus(1, 2)</code>	-	wirus zaraża serwer 1, zakłócając pracę serwerów 1, 2, 3 i 5, które są w odległości co najwyżej 2 od niego
<code>ryzyko(1, 3)</code>	3	zakłócana jest praca serwerów 1, 2 i 3
<code>antywirus(1)</code>	-	program antywirusowy usuwa wirusa z serwera 1 (praca serwerów 1 i 3 przestaje być zakłócana)
<code>ryzyko(1, 3)</code>	1	praca serwera 2 jest nadal zakłócana przez wirusa z serwera 5
<code>wirus(5, 0)</code>	-	nowy wirus zaraża serwer 5 (poprzedni wirus tam pozostaje, nadal zakłócając pracę serwera 2)
<code>ryzyko(1, 3)</code>	1	zakłócana jest praca serwera 2
<code>ryzyko(3, 4)</code>	0	serwery 3 i 4 są połączone bezpośrednim połączeniem i żaden z nich nie jest pod wpływem wirusa

Eksperymenty

Aby ułatwić Ci testowanie swojego rozwiązania, w pliku `siegrader.cpp` udostępniamy przykładowy program oceniający. Aby użyć tego programu, powinieneś umieścić swoje rozwiązanie w tym samym folderze, w pliku `sie.cpp`.

Początkowo w plikach tych możesz znaleźć przykładowe **błędne** rozwiązanie zadania. Dostarczony jest też przykładowy plik `makefile` generujący plik wykonywalny `sieCPP.e` z pliku `sie.cpp` za pomocą komendy:

```
make
```

Powstały w ten sposób plik wykonywalny wczytuje listę nazw funkcji oraz argumentów ze standardowego wejścia, wywołuje odpowiadające im funkcje i wypisuje liczby zwracane przez funkcję `ryzyko` na standardowe wyjście, po jednej w wierszu. Lista na wejściu powinna być w następującym formacie. Pierwszy wiersz powinien zawierać liczbę wywołań m . Każdy z kolejnych m wierszy powinien zaczynać się od znaku `i`, `p`, `w`, `a` lub `r`; wyznacza on funkcję, która ma zostać wywołana: są to odpowiednio funkcje `inicjuj`, `polaczenie`, `wirus`, `antywirus` i `ryzyko`. Następnie, w każdym z tych wierszy występuje jedna lub dwie (w zależności od wybranej funkcji) nieujemne liczby całkowite, które są argumentami wywoływanej funkcji. Zwróć uwagę, że załączony program oceniający nie sprawdza, czy wejście jest poprawnie sformatowane, ani czy spełnia wymagania podane w rozdziałach Komunikacja oraz Ocenianie.

Plik `sie0.in` opisuje przykładowy ciąg wywołań przedstawiony powyżej, natomiast `sie0.out` odpowiada mu ciąg poprawnych wyników.

Ocenianie

Niech q oznacza łączną liczbę wywołań `wirus`, `antywirus` i `ryzyko`. We wszystkich podzadaniach zachodzi $1 \leq n \leq 1\,000\,000$ i $1 \leq q \leq 100\,000$. Ponadto, jak już podano wcześniej, zasięg wirusów oraz długość ścieżki między serwerami w funkcji `ryzyko` wynosi nie więcej niż 500.

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

Podzadanie	Ograniczenia	Punkty
1	$n, q \leq 1000$	20
2	sieć ma topologię gwiazdy: każdy z serwerów $2, \dots, n$ jest bezpośrednio połączony z serwerem numer 1	15
3	tylko jedno zapytanie <code>ryzyko</code>	15
4	brak zapytań <code>antywirus</code>	20
5	brak dodatkowych ograniczeń	30

Testy „ocen”:

- 1ocen:** 25 serwerów, 7 ustawionych w topologię gwiazdy, każdy z zewnętrznych ma doczepione 3 dodatkowe serwery;
- 2ocen:** 100 serwerów; sieć ma topologię gwiazdy;
- 3ocen:** 200 serwerów, wszystkie na jednej ścieżce.