

НАЦИОНАЛНО ОНЛАЙН СЪСТЕЗАНИЕ ПО ИНФОРМАТИКА

„Д-р Младен Манев“

10 май 2020 г.

Задача А4. КОМПЮТЪРНА МРЕЖА

Пешо реши да стане бял хакер. Шефът на компанията, в която той кандидатства за работа, реши лично да се увери в неговите хакерски умения и затова му постави следната задача. Компютърната мрежа на компанията се състои от N на брой сървъра, номерирани с първите N естествени числа. Между някои двойки компютри u и v съществуват двупосочни криптирани връзки. За всяка такава връзка е известно нивото η на сигурност s , което е естествено число не по-голямо от 1 000 000. Освен това знаем, че мрежата е свързана – от всеки сървър чрез една или повече връзки може да се осъществи комуникация с всеки друг. Пешо трябва да изчисли общата сигурност на компютърната мрежа, която се определя като минималната сума от нивата на сигурност на част от криптираните връзки, които осигуряват свързаност на мрежата.

За тази цел Пешо може да задава въпроси, състоящи се от две непресичащи се множества от сървъри A и B , на които шефът му отговаря с минималната по ниво на сигурност връзка между компютрите с номера u и v , такива че u принадлежи на A , а v – на множеството B , или казва, че такава връзка не съществува. При равенство шефът избира произволно някоя от връзките с минимално ниво на сигурност. Тъй като шефът не обича да му губят времето, е наложил ограничението сумата от големините на множествата A за всички въпроси да не надвишава **10 000**, а тази от множествата B – да не надвишава **1 000 000**. Помогнете на Пешо да получи желаната работа.

Задача:

Напишете функция `play()`, която ще се компилира с програмата на журито и ще комуникира с нея, задавайки въпроси от гореописания вид. В края на изпълнението си тя трябва да е открила общата сигурност на компютърната мрежа.

Детайли по имплементацията:

Функцията `void play(int n)`, която трябва да напишете, ще бъде извикана само веднъж от програмата на журито и като аргумент ще получи цялото число N .

За комуникация с програмата на журито Ви се предоставят следните две функции:

```
vector<int> get_connection(const vector<int> &a, const vector<int> &b);  
void submit_security(int t);
```

При всяко извикване на функцията `get_connection`, тя ще връща вектор от три елемента – краищата на търсената криптирана връзка и нейното ниво на сигурност. Ако между нито една двойка върхове от двете множествата не съществува връзка, елементите ще имат стойност 0. Векторите, които предавате като аргументи на функцията, трябва да отговарят на следните условия: да се състоят от положителен брой елементи, сумарно не по-голям от N ; да не съдържат елементи с равни стойности; да съдържат само стойности, които са валидни номера на сървъри. Ако в произволен момент нарушите някое от описаните условия, изпълнението на вашата функция ще бъде прекратено и няма да получите точки за съответния тест.

НАЦИОНАЛНО ОНЛАЙН СЪСТЕЗАНИЕ ПО ИНФОРМАТИКА

„Д-р Младен Манев“

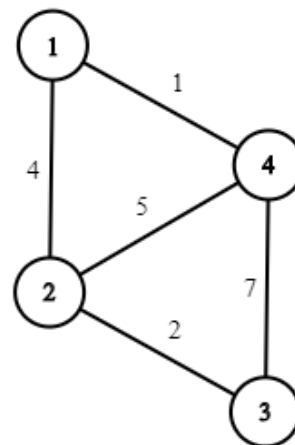
10 май 2020 г.

След като откриете общата сигурност на компютърната мрежа, вашата функция ще трябва да извика функцията `submit_security` и да предаде като аргумент цяло число t , равно на търсената стойност. След това, изпълнението на вашата функция ще бъде прекратено.

Към системата изпратете файл **network.cpp**. В него, освен функцията `play`, може да има всякакви помощни функции, структури, променливи и т. н. Той не трябва да съдържа функция `main()` и задължително трябва да включва хедър файла `network.h` чрез указание към предпроцесора `#include "network.h"` в началото си.

Пример:

Функция на участника	Програма на журито
	<code>play(4)</code>
<code>get_connection({1, 2}, {3, 4})</code>	<code>{1, 4, 1}</code>
<code>get_connection({1, 4}, {2, 3})</code>	<code>{1, 2, 4}</code>
<code>get_connection({3}, {1, 2, 4})</code>	<code>{2, 3, 2}</code>
<code>get_connection({1}, {3})</code>	<code>{0, 0, 0}</code>
<code>get_connection({2}, {4})</code>	<code>{2, 4, 5}</code>
<code>get_connection({3}, {4})</code>	<code>{3, 4, 7}</code>
<code>submit_security(7)</code>	



Обяснение на примера:

Връзките между двойките сървъри (1; 4), (2; 3) и (1; 2) са достатъчни за запазването на свързаността на мрежата. Сумата от нивата им на сигурност е 7, което е и общата сигурност на мрежата.

Ограничения:

$$1 < N \leq 1000$$

Оценяване:

За всеки тест, в който вашата функция правилно е пресметнала общата сигурност на мрежата в рамките на допустимия времеви лимит, ще получите $\min(1, \frac{10001}{s_A+1}) * \min(1, \frac{1000001}{s_B+1}) * 100\%$ от предвидените точки за съответния тест, където s_A и s_B са съответно сумите от големините на множествата A и B от всички въпроси, зададени от функцията `Vi` към програмата на журито.

Локално тестване:

Предоставен Ви е файлът `Lgrader.cpp`, който може да компилирате заедно с вашата функция, за да я тествате. При стартиране трябва да въведете N и M – съответно броя сървъри и връзки, следвани от M тройки числа, описващи сървърите, които свързва поредната връзка, и нейната сигурност. Програмата ще отпечата комуникацията, която се извършва. Може да модифицирате този файл, както искате.