



BULGARIAN

Day 2 Task 4: Saveit

Куриерската фирма Xedef пренася по въздуха пратки между няколко града. В някои от тези градове фирмата извършва специални обработки и тези градове се наричат *разпределители*. Всеки от самолетите на Xedef извършва полети между някои два от градовете, като пренася пратки в двете посоки, според исканията на клиентите.

Пренасянето на една пратка от града на подателя до града на получателя става в няколко стъпки, като на всяка стъпка пратката се пренася от един град в друг със самолета, летящ между тези два града. Нещо повече, последователността от градове, през които ще премине пратката трябва да съдържа поне един град с разпределител на Xedef.

За да улесни процеса, компанията иска да кодира дължината на най-късата последователност от стъпки от всеки град до всеки разпределител и да поставя тази информация като етикет на пратката. (Дължината на най-късата последователност от стъпки от един разпределител до самия него е нула.) Разбира се, кодирането трябва да е компактно.

Напишете две функции **encode(N,H,P,A,B)** и **decode(N,H)**, където **N** е броят на градовете (номерирани от 0 до $N - 1$), а **H** е броят на разпределителите. **P** е броят на двойките от градове, между които има полет. **A** и **B** са такива масиви, че първата двойка свързани с полет градове е $(A[0],B[0])$, втората – $(A[1],B[1])$ и т.н. Градовете, в които има разпределители на фирмата, са с номера от 0 до $H - 1$ ($N \leq 1000$, $H \leq 36$). Всичките **P** зададени (ненаредени) двойки градове са различни.

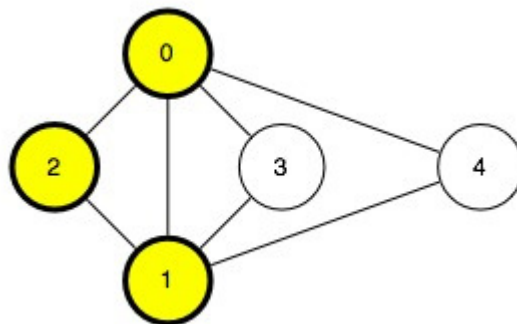
Функцията **encode** трябва да създаде битов низ, от който функцията **decode** да може да определи броя на стъпките между всеки град и всеки разпределител. Функцията **encode** предава низа в грейдера, чрез последователни извиквания на функцията **encode_bit(b)**, където **b** е 0 или 1. Функцията **decode** получава низа от грейдера, чрез последователни извиквания на функцията **decode_bit**, която връща битовете в реда, в който **encode_bit(b)** ги е изпратила към грейдера. Забележете, че броят на извикванията на **decode_bit** не бива да надхвърля броя на извикванията на **encode_bit(b)**.

След декодирането на низа **decode** трябва, за всеки разпределител **h** и всеки град **c** (включително всеки разпределител, т.е. $c=h$), да извика **hops(h,c,d)**, където **d** е минималният брой стъпки, необходими за пренасяне на пратка от **c** до **h**. Т.е. трябва да бъдат направени $H \cdot C$ извиквания на **hops(h,c,d)**. Редът на тези извиквания е без значение. Гарантирано е, че пренасянето на пратка от всеки град до всеки разпределител е възможно.

Забележка. encode и decode трябва да комуникират само през специфицирания по-горе интерфейс. Използване на видими от двете функции (shared) променливи, предаване на данни чрез файл или по мрежата са забранени. В програми на C и C++ може да декларирате променливи като static, за да съхранявате информация за encode или decode, но не трябва да са shared.

Пример

Разгледайте фигурата вдясно. Показани са 5 града ($N=5$), свързани със 7 полета ($P=7$). Градовете 0, 1 и 2 са разпределители ($H=3$). Една стъпка е необходима за да се пренесе пакет между разпределителя 0 и град 3, докато за пренасяне на пратка от разпределителя 2 до град 3 ще са необходими 2 стъпки. Данните за този пример се намират във файла `grader.in.1`. Всички търсени стойности за d са дадени в следната таблица:



d		c				
		0	1	2	3	4
h	0	0	1	1	1	1
	1	1	0	1	1	1
	2	1	1	0	2	2

Подзадача 1 [25 точки]

Функцията `encode` може да направи максимум 16 000 000 извиквания на `encode_bit(b)`.

Подзадача 2 [25 точки]

Функцията `encode` може да направи максимум 360 000 извиквания на `encode_bit(b)`.

Подзадача 3 [25 точки]

Функцията `encode` може да направи максимум 80 000 извиквания на `encode_bit(b)`.

Подзадача 4 [25 точки]

Функцията `encode` може да направи максимум 70 000 извиквания на `encode_bit(b)`.

За **Детайли на имплементацията** виж английския текст.