



Упражнения: Алгоритми в графи

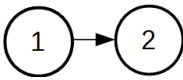
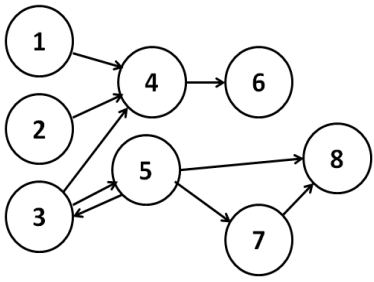
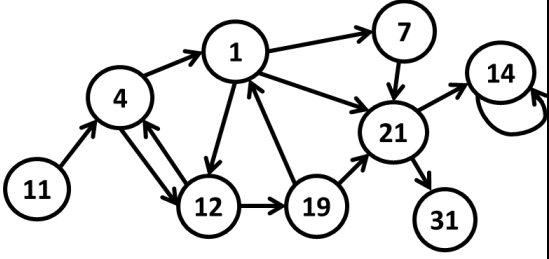
Задача 1. Разстояние между върховете

Даден е ориентиран граф и съответните свързващи двойки върхове. Да се намери най-краткото разстояние между двойките върхове или -1, ако няма път, който ги свързва.

На първия ред се въвежда N - броя на върховете на графа.

На втория ред се въвежда едно цяло число P - броя на двойките върхове, между които ще се търси минималното разстояние. На следващите N реда върховете на графа, а на следващите P реда - двойките върхове.

Пример:

Вход	Визуализация	Изход
2 2 1:2 2: 1-2 2-1		{1, 2} -> 1 {2, 1} -> -1
8 4 1:4 2:4 3:4 5 4:6 5:3 7 8 6: 7:8 8: 1-6 1-5 5-6 5-8		{1, 6} -> 2 {1, 5} -> -1 {5, 6} -> 3 {5, 8} -> 1
9 8 11:4 4:12 1 1:12 21 7 7:21 12:4 19		{11, 7} -> 3 {11, 21} -> 3 {21, 4} -> -1 {19, 14} -> 2 {1, 4} -> 2 {1, 11} -> -1 {31, 21} -> -1



19:1 21 21:14 31 14:14 31: 11-7 11-21 21-4 19-14 1-4 1-11 31-21 11-14		{11, 14} -> 4
--	--	---------------

Подсказки:

За всяка двойка използвайте BFS, за да намерите всички пътища от началния до крайния връх.

Задача 2. Области в матрица

Дадена е матрица с размери NxM, чиито стойности са малки латински букви. Две клетки са съседни, ако те имат обща стена. Напишете програма, която намира свързаните области на съседните клетки, съдържащи една и съща буква. Програмата да извежда общия брой области и броя на зоните за всяка латинска буква, подредени лексикографски.

На първия ред се посочва броят на редовете.

Пример:

Вход	Визуализация	Исход																																																
6 aaccsaac baaaaccc baabaccc bbdaaccc ccdcscsc ccdcscsc	<table><tr><td>a</td><td>a</td><td>c</td><td>c</td><td>c</td><td>a</td><td>a</td><td>c</td></tr><tr><td>b</td><td>a</td><td>a</td><td>a</td><td>a</td><td>c</td><td>c</td><td>c</td></tr><tr><td>b</td><td>a</td><td>a</td><td>b</td><td>a</td><td>c</td><td>c</td><td>c</td></tr><tr><td>b</td><td>b</td><td>d</td><td>a</td><td>a</td><td>c</td><td>c</td><td>c</td></tr><tr><td>c</td><td>c</td><td>d</td><td>c</td><td>c</td><td>c</td><td>c</td><td>c</td></tr><tr><td>c</td><td>c</td><td>d</td><td>c</td><td>c</td><td>c</td><td>c</td><td>c</td></tr></table>	a	a	c	c	c	a	a	c	b	a	a	a	a	c	c	c	b	a	a	b	a	c	c	c	b	b	d	a	a	c	c	c	c	c	d	c	c	c	c	c	c	c	d	c	c	c	c	c	Области: 8 Буква 'a' -> 2 Буква 'b' -> 2 Буква 'c' -> 3 Буква 'd' -> 1
a	a	c	c	c	a	a	c																																											
b	a	a	a	a	c	c	c																																											
b	a	a	b	a	c	c	c																																											
b	b	d	a	a	c	c	c																																											
c	c	d	c	c	c	c	c																																											
c	c	d	c	c	c	c	c																																											
3 aaa aaa aaa	<table><tr><td>a</td><td>a</td><td>a</td></tr><tr><td>a</td><td>a</td><td>a</td></tr><tr><td>a</td><td>a</td><td>a</td></tr></table>	a	a	a	a	a	a	a	a	a	Области: 1 Буква 'a' -> 1																																							
a	a	a																																																
a	a	a																																																
a	a	a																																																



5																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Подсказки:

В началото маркирайте всички клетки като непосетени. Стартирайте рекурсивен DFS (или BFS) от всяка непосетена клетка и маркирайте всички достигнати клетки като посетени. Всяко DFS (или BFS) преминаване ще намери една от свързаните области.

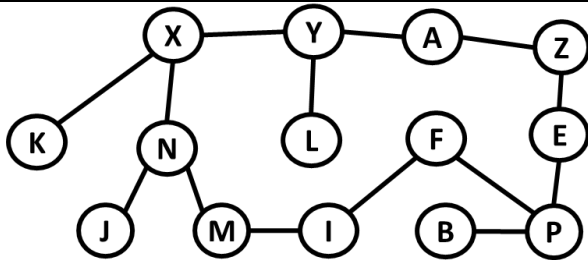
Задача 3. Цикъл в граф

Напишете програма, която проверява дали даден неориентиран граф съдържа цикли или е ацикличен.

Пример:

Input	Picture	Output
C-G		Ациклична: Yes
A-F F-D D-A		Ациклична: No
E-Q Q-P P-B		Ациклична: Yes
K-J J-N N-L N-M M-I		Ациклична: Yes



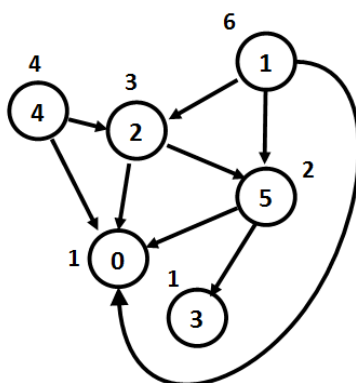
K-X		Ациклична: No
X-Y		
X-N		
N-J		
M-N		
A-Z		
B-P		
I-F		
A-Y		
Y-L		
M-I		
F-P		
Z-E		
P-E		

Подсказки:

Модифицирайте алгоритъма за топологично сортиране.

Задача 4, Заплата

Между служителите в фирма X има йерархия. Служителите могат да имат един или няколко преки началника. Хората, които не управляват никой, се наричат редовни служители и заплатата им е 1. Хората, които управляват поне един служител, се наричат мениджъри. Всеки мениджър взема заплатата, равна на сумата от заплатите на своите пряко управлявани служители. Мениджърите не могат управляват пряко или косвено себе си. Някои служители може да нямат мениджър (като големия шеф). Вижте примерна йерархия във фирма, заедно с изчислените заплати, следвайки гореописаното правило:



И така: служителите 0 и 3 са редовни служители и взимат заплата 1. Всички останали са мениджъри и вземат сумата от заплатите на своите пряко управлявани служители. Например, мениджър 1 взема заплата $3 + 2 + 1 = 6$ (сума от заплатите на служителите 2, 5 и 0). Служителите 4 и 1 нямат мениджър.

В задачата се знаят броя на служителите - N на брой, те са с индекси от 0 до $N - 1$. За всеки служител има информация във вид на низ от символите „Y“ или „N“ в зависимост от това дали настоящият служител е пряк мениджър на служителя i или не.



Съвет: намерете връх, който няма родител и стартирайте DFS, за да изчислите заплатите по дървото рекурсивно.

Вход:

- Входните данни се въвеждат в конзолата
- На първия ред се въвежда стойността на N - цяло число
- На следващите N реда се въвеждат низове от двата символа 'Y' или 'N' с дължина N .

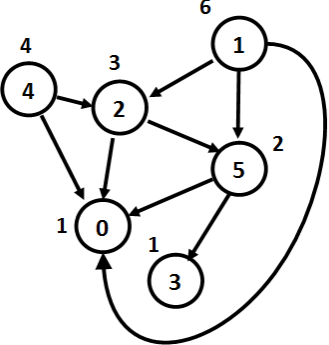
Изход:

- Изходните данни се извеждат в конзолата на един ред.
- На изхода се извежда едно число - сумата от заплатите на всички служители

Ограничения:

- N - цяло число в интервала $[1 \dots 50]$.
- За всеки i -ти ред, i -тия символ е 'N'.
- Ако служител A е мениджър на служител B , B не може да бъде мениджър на A .
- Време за работа на програмата (time limit): 0.1 секунди. Използвана памет:(memory limit) 16 MB.

Пример:

Вход	Изход	Пояснения
1 N	1	Само 1 служител със заплата 1.
4 NNYN NNYN NNNN NYYN	5	Имаме 4 служителя. 0, 1, и 3 са мениджъри на 2. 3 е мениджър на 1. Тогава: заплата(2) = 1 заплата(0) = заплата(2) = 1 заплата(1) = заплата(2) = 1 заплата(3) = заплата(2) + заплата(1) = 2
6 NNNNN N YNYNNY YNNNNY NNNNN N YNYNNN YNNYNN	17	

Задача 5. Прекъсване на цикли

Даден е неориентиран мултиграф. Задачата да отстраните минимален брой ръбове, така, че да направите графа ацикличен. (т.е. Трябва да прекъснете всички цикли). На изхода е необходимо да отпечатате броя на отстранените



ръбове и кои точно са те. Ако няколко ръба могат да бъдат премахнати, за да се прекъсне определен цикъл, премахнете най-малкия от тях по азбучен ред (най-малкият начален връх и най-малкият краен връх по лексикографска наредба).

Пример:

Вход	Визуализация	Изход	Визуализация на графа след отстраняването на ребрата
1 -> 2 5 4 2 -> 1 3 3 -> 2 5 4 -> 1 5 -> 1 3 6 -> 7 8 7 -> 6 8 8 -> 6 7		Брой на премахнатите ребра: 2 1 - 2 6 - 7	
K -> X J J -> K N N -> J X L M X -> K N Y M -> N I Y -> X L L -> N I Y I -> M L A -> Z Z Z Z -> A A A F -> E B P E -> F P P -> B F E B -> F P		Брой на премахнатите ребра: 7 A - Z A - Z B - F E - F I - L J - K L - N	

Подсказки

- Избройте ребрата {s, e} по лексикографска наредба. За всяко ребро {s, e} проверете дали то затваря някакъв цикъл. Ако отговорът е да - премахнете го.
- За да проверите дали дадено ребро {s, e} затваря някакъв цикъл, временно премахнете реброто {s, e} и след това опитайте да намерите път от s до e, използвайки DFS или BFS.