

Софтуниада 2024

Задача 08. Мениджър на пакети

Нает си да напишеш мениджър за пакети, който ще зарежда пакетите в един софтуерен проект.

Дават ни се **N** пакета използвани в един софтуерен проект, както и **M** връзки между тези пакети, където всяка връзка дава информация за зависимост на един пакет от друг пакет (Примерно връзката **1 3**, дава информация че пакета **1** е **зависимост на** пакета **3**). Целта ни е да намерим **подредба на пакетите** така че **всеки пакет да се зареди след като са заредени всичките му зависимости**.

Ограничения

- Броят на пакетите **N** ще е цяло число в диапазона **[1...100 000]**
- Пакетите винаги ще са стойностите от **[0...N-1]**
- Един пакет може да има **0 или повече зависимости**, както и да е **зависим от 0 или повече пакети**
- Броят на връзките **M** ще е цяло число в диапазона **[0...1 000 000]**
- Връзките ще бъдат подадени в произволен ред

Вход

На **първия ред** от конзолата ще получим **броя на пакетите N**

На **вторият ред** от конзолата ще получим **броя на връзките M**

- На **всеки от следващите M реда** ще получим информация за връзка в формата:

{X} {Y}

където **X** пакета е **зависимост** на **Y** пакета.

Изход

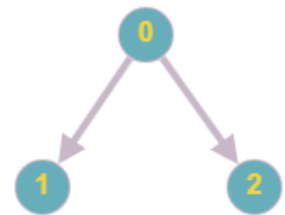
Трябва да изпечатате точно **1 ред** като изход:

- Ако не може да се намери търсената подредба изпечатваме **'circular dependency'**
- Ако такава подредба съществува я изпечатваме в формата:
{1-ви пакет} {2-ри пакет}...{N-ти пакет}
Където **1-ви пакет** е първия пакет, който ще се зареди в проекта, **2-ри пакет** е втория, който ще се зареди и т.н.
 - Ако има много подредби, които спазват условието, изпечатваме тази с **най-малка цифрова стойност**. Примерно подредба **0 1 2** е с по-малка цифрова стойност от подредба **1 0 2** (защото $012 < 102$)

Примерен вход и изход

Вход	Изход
------	-------

3 2 0 1 0 2	0 1 2
JS Вход	Изход
(['3', '2', '0 1', '0 2'])	0 1 2
Коментар	
<p>Гледайки връзките между пакетите, виждаме че пакети 1 и 2 зависят от пакет 0. Имаме 2 варианта как да заредим пакетите в проекта:</p> <p>0 1 2</p> <p>0 2 1</p> <p>Спазвайки условието обаче, избираме подредба 0 1 2, защото има по-малка цифрова стойност, така отговора е:</p> <p>0 1 2</p>	



Вход	Изход
2 2 0 1 1 0	circular dependency
JS Вход	Изход
(['2', '2', '0 1', '1 0'])	circular dependency
Коментар	
<p>Гледайки връзките между пакетите, виждаме че пакет 0 зависи от пакет 1 и пакет 1 зависи от пакет 0. В такава ситуация какъвто и ред да изберем за зареждане ще трябва да заредим пакет преди зависимостите му, така че изпечатваме:</p> <p>circular dependency</p>	



Вход	Изход
------	-------

<pre> 7 5 4 1 5 6 3 0 4 3 1 2 </pre>	<pre> 4 1 2 3 0 5 6 </pre>
JS Вход	Исход
<pre> (['7', '5', '4 1', '5 6', '3 0', '4 3', '1 2']) </pre>	<pre> 4 1 2 3 0 5 6 </pre>
Коментар	
