**Evaluate division**

Equations are given in the format A / B = k, where A and B are variables represented as strings, and k is a real number (floating point number). Given some queries, return the answers. If the answer does not exist, return -1.0.

class Solution **{**

public double**[]** calcEquation**(**String**[][]** equations**,** double**[]** values**,** String**[][]** queries**)** **{**

Map**<**String**,** Map**<**String**,**Double**>>** list **=** **new** HashMap**<>();**

**for(**int i **=** 0**;** i**<**values**.**length**;**i**++){**

list**.**putIfAbsent**(**equations**[**i**][**0**],new** HashMap**<>());** //Se genera el grafo utilizando listas

list**.**putIfAbsent**(**equations**[**i**][**1**],new** HashMap**<>());** //Se usa HashMap

list**.**get**(**equations**[**i**][**0**]).**put**(**equations**[**i**][**1**],**values**[**i**]);** //Se agrega el valor de la ruta

list**.**get**(**equations**[**i**][**1**]).**put**(**equations**[**i**][**0**],**1**/**values**[**i**]);** //Se agrega el inverso del valor

**}**

double **[]** r **=** **new** double**[**queries**.**length**];**

**for(**int i **=** 0**;** i**<** queries**.**length**;**i**++)**

r**[**i**]** **=** dfs**(**queries**[**i**][**0**],**queries**[**i**][**1**],**1**,**list**,new** HashSet**<>());** // Se inicializa con valor 1 por ser el valor de A/A.

**return** r**;**

**}**

double dfs**(**String s**,** String t**,** double r**,** Map**<**String**,** Map**<**String**,**Double**>>** m**,** Set**<**String**>** seen**){**

**if(!**m**.**containsKey**(**s**)** **||** **!**seen**.**add**(**s**))** **return** **-**1**;**

**if(**s**.**equals**(**t**))** **return** r**;**

Map**<**String**,** Double**>** next **=** m**.**get**(**s**);**

**for(**String c **:** next**.**keySet**()){**//Insertar todos los adyacentes.

double result **=** dfs**(**c**,**t**,**r**\***next**.**get**(**c**),**m**,**seen**);** **if(**result **!=** **-**1**)** **return** result**;**

**}**

**return** **-**1**;**

**}**

**}**