

# Exercício Programa 1

MatrizDist.java

Criar um programa em **Java** que leia um grafo na entrada padrão e imprima a **matriz de distâncias** na saída padrão usando a busca em largura.

(**Obs:** Não esqueça de conferir o seu programa usando **todos** os **exemplos** abaixo. Desta forma, você poderá identificar possíveis erros antes da data de entrega. Siga atentamente as **instruções**: **nome** do programa, incluindo maiúsculas e minúsculas, formatação de **entrada** e **saída**, etc. A correção pode ser efetuada por um script, portanto, evite erros desnecessários. Qualquer tipo de erro será descontado.)

## Entrada

A primeira linha contém o total de vértices **N**. As próximas N linhas contém os nomes de cada vértice (numerados de 0 até N-1). A próxima linha contém o total de arcos **M**. As próximas M linhas contém pares de índices de vértices, em que cada par corresponde as pontas inicial e final de cada arco.

## Saída

A matriz de distâncias.

No caso de distância infinita, imprimir um ponto ('.').

## Exemplos

Entrada	Saída
6	0 1 2 2 1 3
1	1 0 1 2 1 3
2	2 1 0 1 2 2
3	2 2 1 0 1 1
4	1 1 2 1 0 2
5	3 3 2 1 2 0
6	
14	
0 1	
0 4	
1 0	
1 2	
1 4	
2 1	
2 3	
3 2	
3 4	
3 5	
4 0	
4 1	
4 3	
5 3	

Entrada	Saída
6	0 1 . 1 2 .
1	. 0 . 2 1 .
2	. 3 0 2 1 1
3	. 1 . 0 2 .
4	. 2 . 1 0 .
5	. . . . . 0
6	
8	
0 1	
0 3	
1 4	
2 4	
2 5	
3 1	
4 3	
5 5	

Entrada	Saída
5	0 1 2 1 2
s	3 0 1 1 2
t	2 3 0 3 1
x	2 1 1 0 1
y	1 2 1 2 0
z	
10	
0 1	
0 3	
1 2	
1 3	
2 4	
3 1	
3 2	
3 4	
4 0	
4 2	

Entrada	Saída
6	0 1 . 1 2 .
u	. 0 . 2 1 .
v	. 3 0 2 1 1
w	. 1 . 0 2 .
x	. 2 . 1 0 .
y	. . . . . 0
z	
8	
0 1	
0 3	
1 4	
2 4	
2 5	
3 1	
4 3	
5 5	

Entrada	Saída
6	0 1 1 1 1 1
0	. 0 3 1 1 2
1	. 1 0 2 2 3
2	. 2 2 0 1 1
3	. 2 2 3 0 1
4	. 1 1 2 2 0
5	
13	
0 1	
0 2	
0 3	
0 4	
0 5	
1 3	
1 4	
2 1	
3 4	
3 5	
4 5	
5 1	
5 2	

Entrada	Saída
8	0 . . . 1 2 2 1
s	2 0 1 1 2 3 4 3
t	2 1 0 1 2 3 4 3
u	1 . . 0 1 2 3 2
v	. . . . 0 1 3 2
w	. . . . 2 0 2 1
x	. . . . 3 1 0 2
y	. . . . 1 2 1 0
z	
13	
0 4	
0 7	
1 2	
1 3	
2 1	
2 3	
3 0	
3 4	
4 5	
5 7	
6 5	
7 4	
7 6	

Entrada	Saída
7	0 1 3 2 1 . 2
a	. 0 . . . . 1
b	3 1 0 2 1 . 1
c	1 2 1 0 2 . 2
d	2 1 2 1 0 . 2
e	1 2 2 1 2 0 3
s	. . . . . 0
t	
12	
0 1	
0 4	
1 6	
2 1	
2 4	
2 6	
3 0	
3 2	
4 1	
4 3	
5 0	
5 3	

Entrada	Saída
7	0 1 2 2 1 . 2
a	. 0 2 2 1 . 1
b	. . 0 . . . 1
c	. . 1 0 . . 2
d	. . 1 1 0 . 1
e	1 2 2 1 2 0 3
s	. . . . . 0
t	
11	
0 1	
0 4	
1 4	
1 6	
2 6	
3 2	
4 2	
4 3	
4 6	
5 0	
5 3	

Entrada	Saída
9	0 1 2 2 2 1 2 . 3
a	. 0 1 1 . . . . 2
b	. . 0 . . . . . 1
c	. . 1 0 . . . . 1
d	. . 2 1 0 . . . 2
e	. 2 3 2 1 0 1 . 3
f	. 1 2 2 1 . 0 . 3
g	1 2 3 3 2 1 2 0 4
s	. . . . . . . 0
t	
14	
0 1	
0 5	
1 2	
1 3	
2 8	
3 2	
3 8	
4 3	
5 4	
5 6	
6 1	
6 4	
7 0	
7 5	

Entrada	Saída
10	0 1 1 2 . 2 . . . 2
a	. 0 . 1 . . . . 1
b	. . 0 1 . 1 . . . 2
c	. . . 0 . . . . 1
d	. . 1 2 0 1 . . . 2
e	. . . . . 0 . . . 1
f	. . 2 3 1 2 0 1 . 3
g	. . . . . 1 . 0 . 2
h	1 2 1 2 2 2 1 2 0 3
s	. . . . . . . . 0
t	
16	
0 1	
0 2	
1 3	
1 9	
2 3	
2 5	
3 9	
4 2	
4 5	
5 9	
6 4	
6 7	
7 5	
8 0	
8 2	
8 6	



Entrada	Saída
7	0 1 2 1 3 . 2
a	3 0 1 4 2 . 1
b	2 3 0 3 1 . 1
c	3 1 1 0 2 . 2
d	1 2 3 2 0 . 3
e	1 2 2 1 1 0 3
s	. . . . . 0
t	
12	
0 1	
0 3	
1 2	
1 6	
2 4	
2 6	
3 1	
3 2	
4 0	
5 0	
5 3	
5 4	

Entrada	Saída
7	0 1 2 4 3 . 2
a	3 0 1 3 2 . 1
b	2 2 0 2 1 . 1
c	3 1 1 0 2 . 2
d	1 1 2 1 0 . 2
e	1 2 2 1 1 0 3
s	. . . . . 0
t	
13	
0 1	
1 2	
1 6	
2 4	
2 6	
3 1	
3 2	
4 0	
4 1	
4 3	
5 0	
5 3	
5 4	

Entrada	Saída
8	0 1 . 1 1 . . 2
a	. 0 . 1 . . . 1
b	. 1 0 1 . . . 2
c	. . . 0 . . . 1
d	. . . . 0 . . 1
e	. 2 1 2 . 0 . 3
f	1 2 1 2 2 1 0 3
s	. . . . . . . 0
t	
13	
0 1	
0 3	
0 4	
1 3	
1 7	
2 1	
2 3	
3 7	
4 7	
5 2	
6 0	
6 2	
6 5	