

# Exercícios

Implemente uma classe em Java que represente um grafo simples e não direcionado utilizando uma lista de adjacências com a estrutura `LinkedList<Integer>` para armazenar os vizinhos de cada vértice. O programa deve ler o grafo da entrada padrão e, para cada vértice  $v$ , produzir:

- O conjunto de vizinhos  $N(v)$
- O grau de  $v$

## Entrada

- Um número inteiro  $n$  representando o número de vértices (numerados de 0 a  $n - 1$ ).
- Um número inteiro  $m$  representando o número de arestas.
- Em seguida,  $m$  linhas com dois inteiros  $u$  e  $v$ , indicando que existe uma aresta entre os vértices  $u$  e  $v$ .

## Saída

Para cada vértice  $v$ , o programa deve imprimir os vizinhos  $N(v)$  e o grau do vértice, no formato:

$N(v) = [\dots]$ , grau =  $k$

## Exemplo

Entrada:

```
4
4
0 1
0 2
1 2
2 3
```

Saída esperada:

```
N(0) = [1, 2], grau = 2
N(1) = [0, 2], grau = 2
N(2) = [0, 1, 3], grau = 3
N(3) = [2], grau = 1
```

Implemente uma classe em Java para representar um grafo simples e não direcionado, em que cada aresta possui um identificador textual. Utilize um objeto do tipo `Map<String, Set<Integer>>` para armazenar a função de incidência  $\varphi$ , cuja finalidade é associar cada aresta ao conjunto dos dois vértices que ela conecta. O programa deve ler as arestas da entrada padrão e, ao final, exibir a função de incidência  $\varphi$ , considerando que  $\varphi(e) = \{u, v\}$  representa que a aresta  $e$  liga os vértices  $u$  e  $v$ .

## Entrada

- Um número inteiro  $m$  representando a quantidade de arestas.
- Em seguida,  $m$  linhas contendo uma string  $e$  (identificador da aresta) e dois inteiros  $u$  e  $v$ , indicando que a aresta  $e$  conecta os vértices  $u$  e  $v$ .

## Saída

Para cada aresta, exiba a função de incidência no formato:

`phi(e) = {u, v}`

## Exemplo

Entrada:

```
3
e1 0 1
e2 0 2
e3 1 3
```

Saída esperada:

```
phi(e1) = {0, 1}
phi(e2) = {0, 2}
phi(e3) = {1, 3}
```

### **Referências**

BONDY, J. A.; MURTY, U. S. R. **Graph theory**. New York: Springer, 2008. (Graduate Texts in Mathematics, v. 244).

CORMEN, Thomas H. et al. **Introduction to Algorithms**. 3. ed. Cambridge: MIT Press, 2009.

FEOFILOFF, Paulo. **Análise de Algoritmos**. Disponível em: [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/](https://www.ime.usp.br/~pf/analise_de_algoritmos/). Acesso em: março de 2025.

KLEINBERG, Jon; TARDOS, Éva. **Algorithm Design**. Boston: Pearson, 2006.