

# Teoria de Grafos    Trabalho 3

## Enunciado

O trabalho consiste em implementar uma série de funções colocadas no módulo [Grafo2.hs](#), criado por você, cujos enunciados serão dados a seguir.

Crie um arquivo de testes [TesteGrafo2.hs](#) que importe o módulo [Grafo2.hs](#) para testar cada uma de suas funções em *pelo menos* três grafos diferentes e que não sejam apenas variações do mesmo grafo. Se desejar, pode importar o módulo feito por você para o trabalho anterior. As funções são:

1. **fundeVértices**  $g\ v\ w$ , devolve um grafo com a fusão de um par de vértices  $v$  e  $w$  no grafo  $g$ . Use o índice do menor dos vértices  $v$  e  $w$  como o vértice resultante da fusão.
2. **contraíVértices**  $g\ v\ w$ , devolve um grafo com a contração de um par de vértices  $v$  e  $w$  no grafo  $g$ . Use o índice do menor dos vértices  $v$  e  $w$  como o vértice resultante da contração.
3. **contraíAresta**  $g\ (v,w)$ , devolve um grafo com a contração da aresta  $(v,w)$ .
4. **éPasseio**  $g\ vs$ , verifica se a lista de vértices  $vs$  é um passeio no grafo  $g$ .
5. **éPasseioAberto**  $g\ vs$ , verifica se a lista de vértices  $vs$  é um passeio aberto no grafo  $g$ .
6. **éPasseioFechado**  $g\ vs$ , verifica se a lista de vértices  $vs$  é um passeio fechado no grafo  $g$ .
7. **éTrilha**  $g\ vs$ , verifica se a lista de vértices  $vs$  é uma trilha no grafo  $g$ .
8. **éCaminho**  $g\ vs$ , verifica se a lista de vértices  $vs$  é um caminho no grafo  $g$ .
9. **éCiclo**  $g\ vs$ , verifica se a lista de vértices  $vs$  é um ciclo no grafo  $g$ .
10. **éGrafoCíclico**  $c$ , verifica se o grafo  $c$  é cíclico.
11. **éCn**  $g\ n$ , verifica se o grafo  $g$  é cíclico com  $n$  vértices.
12. **éGrafoCaminho**  $g$ , verifica se o grafo  $g$  é um grafo caminho.
13. **éPn**  $p\ n$ , verifica se o grafo  $p$  é um grafo caminho com  $n$  vértices.
14. **éGrafoBipartido**  $g\ v1\ v2$ , verifica se um grafo  $g$  é bipartido com a partição dos vértices dada pelas listas de vértices  $v1$  e  $v2$ .
15. **éTrilhaEuleriana**  $g\ t$ , verifica se a trilha  $t$  é euleriana no grafo  $g$ .
16. **éGrafoEuleriano**  $g$ , verifica se um dado  $g$  é euleriano. Use o Teorema de Euler.

17. **éSemiEuleriano**  $g$ , verifica se um dado  $g$  é semi-euleriano.
18. **éHamiltonianoOre**  $g$ , verifica se um grafo  $g$  é hamiltoniano usando o teorema de Ore.
19. **éHamiltonianoDirac**  $g$ , verifica se um grafo  $g$  é hamiltoniano usando o teorema de Dirac.
20. **fecho**  $g$ , devolve um grafo que é o fecho do grafo  $g$ .
21. **buscaGenérica**  $g$ , devolve uma lista de vértices na sequência em que são primeiramente visitados (marcados) em uma busca genérica no grafo  $g$ .
22. **buscaLargura**  $g$ , devolve uma lista de vértices na sequência em que são primeiramente visitados (marcados) em uma busca em largura no grafo  $g$ .
23. **buscaProfundidade**  $g$ , devolve uma lista de vértices na sequência em que são primeiramente visitados (marcados) em uma busca em profundidade no grafo  $g$ .
24. **menorCaminho**  $g$   $u$   $v$ , devolve uma lista com os vértices no menor caminho em  $g$  entre os vértices  $u$  e  $v$ . Adapte a busca em largura e a use para resolver este problema.