

Teoria de Grafos Trabalho 3

Enunciado

O trabalho consiste em implementar uma série de funções que serão colocadas no módulo `Grafo.hs`, criado por você, cujos enunciados serão dados na sequência. Importe o módulo `BaseGrafo.hs`, disponibilizado no Microsoft Teams.

Crie um arquivo de testes `Teste3.hs` que importe os módulos `Grafo.hs`, deste trabalho, e `GrafosEspeciais.hs`, do trabalho anterior, para testar cada uma dessas funções em pelo menos três grafos diferentes e que não sejam apenas variações do mesmo grafo.

Não use biblioteca alguma que implemente diretamente as funções pedidas.

As funções a serem implementadas são:

1. **éVértIsolado** `g v`, devolve `True` se o vértice `v` é isolado no grafo `g` ou `False`, em caso contrário.
2. **éVértTerminal** `g v`, devolve `True` se o vértice `v` é terminal no grafo `g` ou `False`, em caso contrário.
3. **éVértPar** `g v`, devolve `True` se o vértice `v` é um vértice par no grafo `g` ou `False`, em caso contrário.
4. **éVértImpar** `g v`, devolve `True` se o vértice `v` é um vértice ímpar no grafo `g` é trivial ou `False`, em caso contrário.
5. **sãoArestasAdjs** `g (t,u) (v,w)` devolve `True` se as duas arestas `(t,u)` e `(v,w)` são adjacentes no grafo `g`. Note que arestas tais como `(1,2)`, `(3,2)` e `(2,1)`, `(3,2)` também são adjacentes.
6. **éTrivial** `g`, devolve `True` se o grafo `g` é trivial ou `False`, se `g` não é trivial.
7. **seqGraus** `g` devolve a sequência de graus de `g`.
8. **grauMax** `g`, devolve o grau máximo de `g`.
9. **grauMin** `g`, devolve o grau mínimo de `g`.
10. **éRegular** `g`, devolve `True` se o grafo `g` é regular ou `False`, em caso contrário.
11. **ékRegular** `g k`, devolve `True` se o grafo `g` é `k`-regular ou `False`, em caso contrário.
12. **éVazio** `g`, devolve `True` se o grafo `g` é vazio ou `False`, em caso contrário.
13. **éNulo** `g`, devolve `True` se o grafo `g` é nulo ou `False`, em caso contrário.
14. **éKn** `g n`, devolve `True` se o grafo `g` é completo com `n` vértices ou `False`, em caso contrário.

15. **éCompleto** *g*, devolve **True** se o grafo *g* é completo ou **False**, em caso contrário.
16. **éSubgrafo** *h g*, devolve **True** se o grafo *h* é um subgrafo do grafo *g* ou **False**, em caso contrário.
17. **éSubgrafoPróprio** *h g*, devolve **True** se o grafo *h* é um subgrafo próprio do grafo *h* ou **False**, em caso contrário.
18. **éSubgrafoInduzidoVértices** *h g vh*, devolve **True** se o grafo *h* é um subgrafo induzido por um lista de vértices *vh* do grafo *g* ou **False**, em caso contrário.
19. **éSubgrafoInduzidoArestas** *h g ars*, devolve **True** se o grafo *h* é um subgrafo induzido por um lista de arestas *ars* do grafo *g* ou **False**, em caso contrário.
20. **éClique** *h g*, devolve **True** se o grafo *h* é um clique do grafo *g* ou **False**, em caso contrário.
21. **éCjIndependenteVértices** *h g*, devolve **True** se o grafo *h* é um conjunto independente de vértices do grafo *g* ou **False**, em caso contrário.
22. **insereVértices** *g vs*, devolve um novo grafo com todos os vértices na lista *vs* inseridos em *g*. A função deve ignorar vértices que já existam no grafo *g*.
23. **removeVértices** *g vs*, devolve um grafo com todos os vértices (e arestas incidentes) na lista *vs* removidos de *g*. A função deve ignorar vértices que não estão presentes no grafo *g*. Dica: já existe uma função em **BaseGrafo.hs** que remove um único vértice. Use-a.
24. **insereArestas** *g ars*, devolve um novo grafo com todas as arestas na lista *ars* inseridas em *g*. A função deve ignorar arestas que já existam no grafo *g* e deve indicar erro se algum dos vértices das arestas não estiver em *g*.
25. **removeArestas** *g ars*, devolve um grafo com todas as arestas na lista *ars* removidas de *g*. A função deve ignorar arestas que não estão presentes no grafo *g*. Dica: já existe uma função em **BaseGrafo.hs** que remove uma única aresta. Use-a.
26. **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.
27. **contraíVértices** *g v w*, devolve um grafo com a contração de um par de vértices *v* e *w* no grafo *g*. Use o rótulo do menor dos vértices *v* e *w* como o rótulo do vértice resultante da contração.
28. **contraíAresta** *g (v,w)*, devolve um grafo com a contração da aresta *(v,w)*.