

Teoria de Grafos Trabalho 4

Enunciado

O trabalho consiste em implementar uma série de funções colocadas no módulo [Grafo3.hs](#), criado por você, cujos enunciados serão dados a seguir. Crie um arquivo de testes [Teste-Grafo3.hs](#) que importe o módulo [Grafo3.hs](#) para testar cada uma dessas funções em pelo menos três grafos diferentes e que não sejam apenas variações do mesmo grafo. Se desejar, pode *importar* os módulos feitos por você nos trabalhos anteriores. As funções são:

1. **éConexo g**, devolve [True](#) se o grafo **g** é conexo ou [False](#), em caso contrário.
2. **numCompConexas g** devolve um número natural para o número de componentes conexas de **g**.
3. **ciclo g u**, devolve uma lista de vértices representando um ciclo em **g**. A função inicia a busca a partir do vértice **u**. Caso não haja nenhum ciclo, devolve a lista vazia. Note que o vértice **u** não necessariamente deve pertencer ao ciclo retornado, ele é apenas o ponto de partida da busca.
4. **distância g u v**, devolve a distância entre os vértices **u** e **v** em **g**.
5. **dijkstra g v**, devolve um par (d,p) de vetores contendo em **d** as menores distâncias de **v** até qualquer outro vértice de **g** e em **p** os predecessores. Use o algoritmo de Dijkstra.
6. **excentricidade g v**, devolve a excentricidade de **v** em **g**.
7. **raio g v**, devolve o raio de **g**.
8. **diâmetro g v**, devolve o diâmetro de **g**.
9. **centro g**, devolve uma lista contendo os vértices no centro de **g**.
10. **éArticulação g u**, devolve [True](#) se o vértice **u** é um vértice de corte em **g** ou [False](#), em caso contrário.
11. **éPonte g (u,v)**, devolve [True](#) se a aresta **(u,v)** é uma ponte em **g** ou [False](#), em caso contrário.
12. **conectividade g** devolve um número natural **k** para a conectividade de **g**.
13. **éBiconexo g**, devolve [True](#) se o grafo **g** é biconexo ou [False](#), em caso contrário.
14. **sãoCaminhosDisjVértices g c1 c2**, devolve [True](#) se os caminhos **c1** e **c2** são internamente disjuntos em vértices no grafo **g** ou [False](#), em caso contrário.
15. **trilhaEulerFleury g** devolve uma lista de vértices representando uma trilha euleriana no grafo **g**. Se tal trilha não existir, devolve uma lista vazia. Use o algoritmo de Fleury.

16. **trilhaEulerHierholzer**, g devolve uma lista de vértices representando uma trilha euleriana no grafo g . Se tal trilha não existir, devolve uma lista vazia. Use o algoritmo de Hierholzer.