

ACH2024 Algoritmos e Estruturas de Dados II

Prof. Ivandr  Paraboni ivandre@usp.br

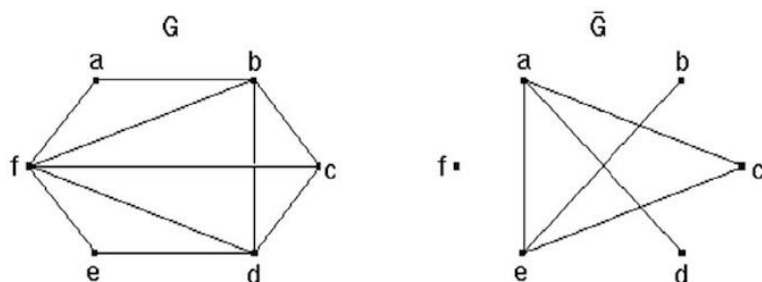
Exerc cios b sicos em grafos

Exceto quando indicado, **todos exerc cios** podem ser resolvidos em **listas ou matrizes** de adjac ncias.

Para verificar se a resposta est  correta ou n o, use o compilador. Esta lista n o disp e de gabarito.

1. Escreva um algoritmo para contar a quantidade de la os em um grafo.
2. Varia  o: remover todos os la os encontrados.
3. Escreva um algoritmo para destruir as arestas de um grafo, tornando-o vazio.
4. Seja um grafo **g** dirigido. Escreva um algoritmo para retornar o grafo transposto de **g**.
5. Escreva um algoritmo que dado um grafo **m** representado em matriz, retorne o mesmo grafo em listas de adjac ncias.
6. Uma  rvore enraizada   um grafo ac clico, conexo e dirigido, com um  nico v rtice fonte de onde todas as arestas partem. Escreva um algoritmo que, dado um grafo **g**, verifique se   uma  rvore enraizada ou n o, retornando *true/false* conforme o caso.
7. Seja um grafo **g** n o-dirigido ponderado (com um peso inteiro associado a cada aresta). Escreva um algoritmo que, dado **g** e um custo m nimo *int c*, retorne uma c pia de **g** contendo apenas as arestas de custo maior do que *c*.
8. Sejam dois grafos **g1** e **g2** contendo exatamente os mesmos v rtices. Verifique se **g2**   um subgrafo de **g1**, retornando *true/false* conforme o caso. Para tornar o problema mais interessante, considere que um grafo   representado em listas e outro em matriz.
9. Dados dois grafos **g1** e **g2**, escreva um algoritmo que retorne um grafo **g3** cujas arestas estejam presentes em **g1** mas n o em **g2**.
10. Considere um grafo dirigido em lista de adjac ncias representando uma rede de troca de emails, onde cada v rtice representa um usu rio, e cada aresta representa um email enviado. Cada aresta possui um campo *int id* que representa o conte do da mensagem (ou seja, o campo *id* faz parte da defini  o do n  das listas de adjac ncias). Problema: circula pela rede uma mensagem **x** do tipo spam. Escreva um algoritmo para exibir os usu rios suspeitos de iniciar a propaga  o de **x**. Um usu rio   considerado suspeito se ele pr prio n o recebeu a mensagem **x** nenhuma vez.
11. Considere um grafo dirigido em lista de adjac ncias representando uma rede de chamadas telef nicas entre diversas unidades de uma empresa, onde cada v rtice representa uma unidade, e cada aresta representa uma chamada efetuada. As unidades podem estar em diferentes pa ses, identificados por um campo *int pa s* definido no respectivo v rtice. Problema: a conta de telefone global da empresa est  muito alta. Escreva um algoritmo que identifique a unidade que efetua chamadas para o maior n mero de pa ses. Havendo empate, retorne qualquer resposta poss vel.
12. Seja um grafo **g** n o-dirigido. Escreva uma fun  o para detectar ciclos em **g**, retornando *true/false*.
13. Varia  o 1: remover as arestas que provocam ciclo.
14. Varia  o 2: retornar o comprimento (i.e., a quantidade de arestas) do maior ciclo encontrado.
15. Seja um grafo **g** n o-conexo e n o-dirigido. Escreva uma fun  o para contar a quantidade de grupos disjuntos de v rtices mutuamente alcan  veis em **g**.
16. Varia  o: ao inv s de contar os grupos, retornar uma lista ligada contendo os v rtices do maior grupo identificado. Se dois ou mais grupos possuem a mesma quantidade de v rtices, retornar a lista de v rtices de qualquer um.
17. Seja um grafo **g** e dois v rtices **a** e **b**. Verifique se h  um caminho qualquer entre **a** e **b** retornando *true/false* conforme o caso.
18. Varia  o 1: contar quantos v rtices h  no caminho de **a** at  **b**.
19. Varia  o 2: retornar a lista dos v rtices que comp e o caminho de **a** at  **b**.
20. Um grafo n o dirigido   completo se todos seus v rtices s o adjacentes, ou seja, existe uma aresta conectando cada par de v rtices de um grafo. Escreva um algoritmo que, dado um grafo simples **g**, verifique se **g**   completo, retornando *true/false* conforme o caso.

21. Seja $G = (V; A)$ um grafo simples e $\bar{G} = (V; V2-A)$ seu complemento, onde $V2$ é o conjunto de todos os pares de vértices em V . Escreva um método que, dado um grafo simples G , retorne seu complemento \bar{G} .



Da esquerda para direita, um grafo e seu complemento.

22. Seja um grafo g representando salas de aula (vértices) e suas ligações (arestas). Cada sala possui uma ocupação representada por um inteiro. Escreva um algoritmo que, a partir da sala atual i , encontre a sala vazia mais próxima, retornando o número do vértice correspondente. Havendo mais de uma sala que atenda estas condições, retorne a primeira que encontrar.
23. Variação: havendo empate, retorne uma lista ligada contendo todas as salas vazias mais próximas.
24. Para todos os vértices de um grafo, calcular o tamanho do caminho mais curto a partir de um vértice inicial i .
25. Seja um grafo não-dirigido representando uma rede social. Os vértices são os usuários e as arestas indicam relações (e.g., de amizade) entre pares de usuários. Dado um usuário i , escreva um algoritmo para exibir todos os usuários relacionados a i com até d graus de distância (medida em quantidade de arestas). Os amigos imediatos estão no grau 1, os amigos dos amigos no grau 2, e assim por diante.
26. Seja um grafo dirigido representando trocas de email entre usuários. Os vértices são usuários e as arestas orientadas indicam que houve envio de mensagens na respectiva direção, incluindo um contador do número de mensagens enviadas. Escreva um algoritmo que, dado um usuário atual i , retorne uma lista ligada contendo todos os usuários que estão diretamente relacionados com i , e que enviaram ou receberam pelo menos k mensagens de/para i . A constante k é fornecida como parâmetro de entrada para a função.
27. Seja um grafo representando uma malha aérea. Vértices são cidades e arestas são voos. Escreva um algoritmo que, dada uma cidade origem a , um destino b e uma companhia aérea c , encontre o trajeto com menor número de conexões de a até b voando apenas pela companhia c . A resposta deve ser fornecida na forma de uma lista ligada de vértices de a até b .
28. Seja um grafo representando as ruas de uma cidade ligando pontos de interesse identificados por um código numérico inteiro (1=hotéis, 2=restaurante etc.) Escreva um algoritmo que, dado um código de ponto de interesse x e uma posição atual i , retorne o vértice contendo o x mais próximo. Havendo empate, retorne o primeiro que encontrar.
29. Variação: considere ainda que existe um local n que *não* deve ser visitado (por exemplo, n pode ser uma área da cidade que foi interditada por alguma razão). Modifique o algoritmo de acordo.