

ACH2024 Algoritmos e Estruturas de Dados II

Prof. Ivandré Paraboni ivandre@usp.br

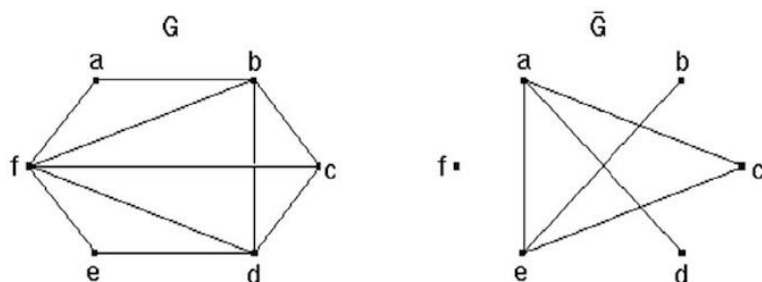
Exercícios básicos em grafos

Exceto quando indicado, **todos exercícios** podem ser resolvidos em **listas ou matrizes** de adjacências.

Para verificar se a resposta está correta ou não, use o compilador. Esta lista não dispõe de gabarito.

1. Escreva um algoritmo para contar a quantidade de laços em um grafo.
2. Variação: remover todos os laços 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 adjacências.
6. Uma árvore enraizada é um grafo acíclico, conexo e dirigido, com um único vértice fonte de onde todas as arestas partem. Escreva um algoritmo que, dado um grafo g , verifique se é uma árvore enraizada ou não, retornando true/false conforme o caso.
7. Seja um grafo g não-dirigido ponderado (com um peso inteiro associado a cada aresta). Escreva um algoritmo que, dado g e um custo mínimo int c , retorne uma cópia de g contendo apenas as arestas de custo maior do que c .
8. Sejam dois grafos g_1 e g_2 contendo exatamente os mesmos vértices. Verifique se g_2 é um subgrafo de g_1 , retornando true/false conforme o caso. Para tornar o problema mais interessante, considere que um grafo é representado em listas e outro em matriz.
9. Dados dois grafos g_1 e g_2 , escreva um algoritmo que retorne um grafo g_3 cujas arestas estejam presentes em g_1 mas não em g_2 .
10. Considere um grafo dirigido em lista de adjacências representando uma rede de troca de emails, onde cada vértice representa um usuário, e cada aresta representa um email enviado. Cada aresta possui um campo int id que representa o conteúdo da mensagem (ou seja, o campo id faz parte da definição do nó das listas de adjacências). Problema: circula pela rede uma mensagem x do tipo spam. Escreva um algoritmo para exibir os usuários suspeitos de iniciar a propagação de x . Um usuário é considerado suspeito se ele próprio não recebeu a mensagem x nenhuma vez.
11. Considere um grafo dirigido em lista de adjacências representando uma rede de chamadas telefônicas entre diversas unidades de uma empresa, onde cada vértice representa uma unidade, e cada aresta representa uma chamada efetuada. As unidades podem estar em diferentes países, identificados por um campo int país definido no respectivo vértice. Problema: a conta de telefone global da empresa está muito alta. Escreva um algoritmo que identifique a unidade que efetua chamadas para o maior número de países. Havendo empate, retorne qualquer resposta possível.
12. Seja um grafo g não-dirigido. Escreva uma função para detectar ciclos em g , retornando true/false.
13. Variação 1: remover as arestas que provocam ciclo.
14. Variação 2: retornar o comprimento (i.e., a quantidade de arestas) do maior ciclo encontrado.
15. Seja um grafo g não-conexo e não-dirigido. Escreva uma função para contar a quantidade de grupos disjuntos de vértices mutuamente alcançáveis em g .
16. Variação: ao invés de contar os grupos, retornar uma lista ligada contendo os vértices do maior grupo identificado. Se dois ou mais grupos possuem a mesma quantidade de vértices, retornar a lista de vértices de qualquer um.
17. Seja um grafo g e dois vértices a e b . Verifique se há um caminho qualquer entre a e b retornando true/false conforme o caso.
18. Variação 1: contar quantos vértices há no caminho de a até b .
19. Variação 2: retornar a lista dos vértices que compõe o caminho de a até b .
20. Um grafo não dirigido é completo se todos seus vértices são adjacentes, ou seja, existe uma aresta conectando cada par de vértices de um grafo. Escreva um algoritmo que, dado um grafo simples g , verifique se g é 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.