Buscas em grafos

Busca em largura

Questão 1. (CLRS) (2ed) Exercícios: 22.2-1, 22.2-2, 22.2-3, 22.2-4, 22.2-5, 22.2-6, 22.2-7(*)

Questão 2. (CLRS) (3ed) Exercícios: 22.2-1, 22.2-2, 22.2-3, 22.2-4, 22.2-5, 22.2-6, 22.2-7, 22.2-8(*)

Questão 3. (CLRS) (3ed, 22.2-3) Show that using a single bit to store each vertex color suffices by arguing that the BFS procedure would produce the same result if lines 5 and 14 were removed.

Questão 4. (CLRS) O diâmetro de uma árvore $T = (V,E)$ é definida como $\max_{u,v \in V} \delta(u,v)$, isso é, o mais longo entre todos os caminhos de distância mínima na árvore. Dê um algoritmo eficiente para calcular o diâmetro de uma árvore e analise o tempo de execução de seu algoritmo.

Questão 5. (Horowitz et al.) Projete um algoritmo para decidir se um dado grafo não direcionado $G = (V,E)$ contém um ciclo de tamanho 4. O tempo de execução de seu algoritmo deve ser $O(V^3)$.

Questão 6. Daniel, que é físico e matemático, acredita ter resolvido um problema importante tanto na Física quanto na Matemática. Temendo as consequências do seu feito, mas querendo preservar sua descoberta, ele encriptou um documento e entregou a chave a um amigo, físico, e a mensagem cifrada uma amiga, matemática. Ele enviou o seguinte e-mail para o físico: *Fiz uma descoberta muito importante! Compartilhe essa chave com todos seus amigos físicos exatamente um dia após recebê-la. Se por um acaso conseguir descobrir o conteúdo, por favor, não divulgue mais nada a mais ninguém!* E escreveu parecido para a matemática: *Fiz uma descoberta muito importante! Compartilhe essa cifra com todos seus amigos matemáticos exatamente um dia após recebê-la. Se por um acaso conseguir descobrir o conteúdo, por favor, não divulgue mais nada a mais ninguém!*

Acontece que ele se esqueceu de um detalhe: como ele, diversas pessoas são físicas e matemáticas. Suponha que você conheça as relações de amizade entre todos os físicos e matemáticos e que todos respeitem a vontade de Daniel. Escreva um algoritmo eficiente que conte o número de pessoas que terão acesso a sua descoberta.

Busca em profundidade

Questão 7. (CLRS) (3ed) Exercícios: 22-3.1 22-3.2 22-3.3 22-3.4 22-3.5 22-3.6 22-3.7 22-3.8 22-3.9 22-3.10 22-3.11 22-3.12

Questão 8. (CLRS) (2ed) Exercícios: 22-3.1 22-3.2 22-3.3 22-3.4 22-3.5 22-3.6 22-3.7 22-3.8 22-3.9 22-3.10 22-3.11

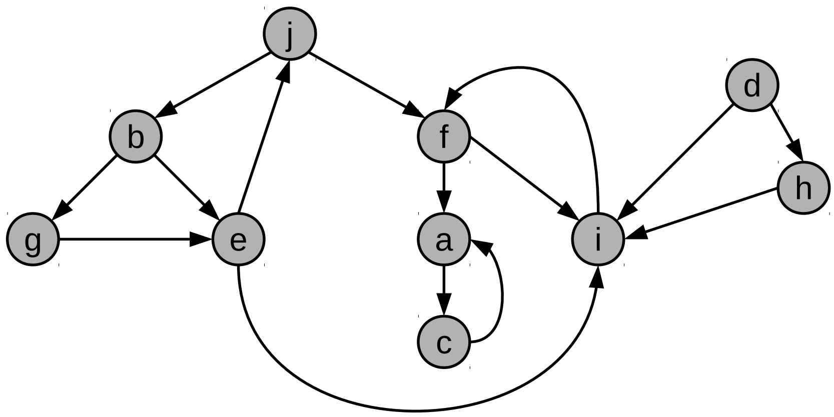
Questão 9. (CLRS) Problemas: 22-3

Questão 10. (Manber) A entrada é um grafo não direcionado $G = (V,E)$, uma árvore geradora T de G e um vértice v. Projete um algoritmo para determinar se T é uma árvore de busca em profundidade enraizada em v. Em outras palavras, determine se T pode ser a saída de DFS quando as arestas estão listadas em alguma ordem e começando pelo vértice v.

Questão 11. Lembre-se de que o algoritmo de busca em profundidade, DFS, pode ser implementado por meio de uma pilha, ao invés de recursão. Suponha que o comprimento máximo de um caminho direcionado em um grafo direcionado G é no máximo R, isso é, todo caminho direcionado de G tem no máximo R arcos. Considere agora *sua* implementação não recursiva de DFS. Prove ou desprove que a pilha P nunca terá mais do que $R+1$ elementos na execução da chamada $\text{DFS}(G)$.

Questão 12. Implemente uma busca em profundidade (DFS) usando uma pilha (de forma a eliminar a recursão). O seu algoritmo deverá *devolver uma floresta de busca em profundidade* representada por um vetor π e deve executar em tempo $O(V+E)$. Você pode utilizar as sub-rotinas de uma pilha como caixas-pretas: CRIARPILHA(Q), TOPO(Q), DESEMPILHAR(Q), EMPILHAR(Q, v).

Questão 13. Mostre como a busca em profundidade funciona no grafo da figura abaixo. Presuma que o laço da rotina principal DFS considera os vértices em ordem alfabética e presuma que cada lista de adjacências está ordenada alfabeticamente. Mostre o tempo de descobrimento (**d**) e o tempo de término (**f**) de cada vértice na caixa abaixo e classifique cada aresta (indique na figura, usando letras F, B, C, T para avanço, retorno, cruzamento e árvore).



	a	b	c	d	e	f	g	h	i	j
d										
f										

Questão 14. Remova as arestas $(b,g),(e,j),(a,c),(f,i)$ do grafo da questão anterior. O grafo obtido é acíclico. Obtenha uma ordenação topológica utilizando o seguinte algoritmo: enquanto houver vértice, remova o sorvedouro com menor letra (na ordem alfabética) e insira em uma lista. Escreva a ordenação topológica obtida.

Ordenação topológica

Questão 15. (CLRS) Exercícios: 22.4-1, 22.4-2, 22.4-3, 22.4-4, 22.4-5,

Questão 16. (CLRS) (22.4-5) Uma outra maneira de obter uma ordenação topológica em um grafo direcionado $G=(V,E)$ é repetidamente encontrar um vértice com grau de entrada 0, incluí-lo na solução e depois remover do grafo esse vértice com todas as arestas de saída. Explique como implementar essa ideia de forma que o algoritmo execute em tempo $O(V+E)$. O que acontece com esse algoritmo se G tiver ciclos?

Componentes fortemente conexas

Questão 17. (CLRS) Exercícios: 22.5-1, 22.5-2, 22.5-3, 22.5-4, 22.5-5, 22.5-6, 22.5-7,

Questão 18. (CLRS) Problemas: 22-2

Questão 19. (CLRS) (22.5-4) Mostre que para cada grafo direcionado G, vale $((G^{\mathrm{T}})^{\mathrm{SCC}})^{\mathrm{T}} = G^{\mathrm{SCC}}$. Isso é, o transposto do grafo de componentes de G^{T} é o mesmo que o grafo de componentes do grafo G.