

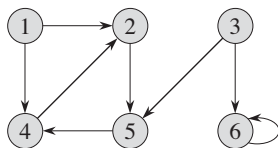
MC202 — Estruturas de Dados
Lista de Exercícios 4
Grafos

Primeiro semestre de 2017 - Turmas B e C
Professor: Emilio Franceschini
franceschini@ic.unicamp.br

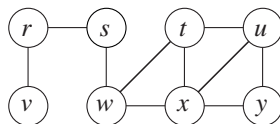
Legenda:

- ★ – Fácil
- ★★ – Médio

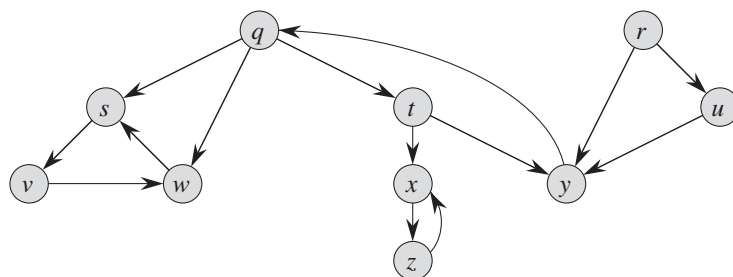
1. ★ Descreva as diferenças entre a representação de grafos através de uma matriz de adjacências e através de listas de adjacências. Simplificadamente descreva quando uma representação é preferível à outra.
2. ★ Escreva um código que recebe um grafo representado por uma matriz de adjacências e que devolve um grafo representado através de listas de adjacências.
3. ★ Escreva um código que recebe um grafo representado por listas de adjacências e que devolve um grafo representado através de uma matriz de adjacências.
4. ★ [CLRS] Dada um grafo dirigido representado por listas de adjacências, quanto tempo (número de operações) leva para determinar o grau de saída de um vértice? Quanto tempo leva para determinar o grau de entrada de um vértice? E se o grafo for representado por uma matriz de adjacências?
5. ★ [CLRS] Mostre um grafo representado por listas de adjacência para uma árvore binária completa com 7 vértices. Mostre a mesma árvore mas desta vez com uma representação baseada em uma matriz de adjacências. Assuma que os vértices são numerados de 1 a 7, 1 sendo a raiz e os demais como se estivessem armazenados em um heap.
6. ★★ [CLRS] Suponha que em lugar de uma lista ligada contendo os arcos (u, v) partindo do vértice u e incidentes em v , nós utilizássemos uma tabela de hash com chave v . Assumindo que a busca por cada uma das arestas é igualmente provável, qual é o tempo esperado para determinar se uma aresta (u, v) está no grafo? Qual é a desvantagem desta abordagem? Sugira uma estrutura de dados alternativa para resolver os problemas apontados. Sua alternativa criar outros problemas que não existiam com uma tabela de hash?
7. ★ [CLRS] Mostre o resultado da execução do algoritmo de busca em largura no grafo representado abaixo. Utilize o nó 3 como ponto de partida. Não esqueça de indicar o estado da fila a cada passo assim como os predecessores e o comprimento do caminho até cada nó.



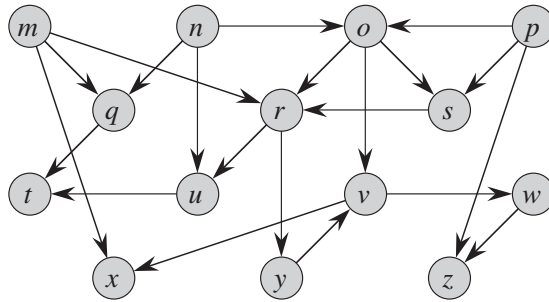
8. ★ [CLRS] Mostre o resultado da execução do algoritmo de busca em largura no grafo não orientado representado abaixo. Utilize o nó u como ponto de partida. Não esqueça de indicar o estado da fila a cada passo assim como os predecessores e o comprimento do caminho até cada nó.



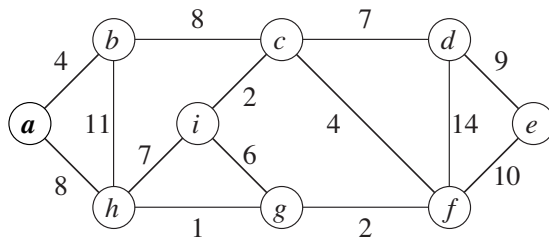
9. ★★ Altere o algoritmo de busca em largura visto em aula para que em lugar de um grafo representado por listas de adjacências ele funcione em um grafo representado por uma matriz de adjacências. Isto altera o tempo de execução do algoritmo?
10. ★ Escreva uma versão recursiva do algoritmo de busca em profundidade descrito em aula.
11. ★ Escreva uma versão do algoritmo de busca em profundidade descrito em aula que receba como entrada um grafo representado por uma matriz de adjacências.
12. ★ Altere o algoritmo iterativo de busca em profundidade visto em aula para que ele marque o instante de descoberta e o instante de término da visita de cada um dos vértices de um grafo orientado. Seu algoritmo deve visitar todos os vértices do grafo, ainda que ele possua mais de um componente, ou seja, mesmo que hajam grupos vértices que não tem arestas conectando uns aos outros.
13. ★ [CLRS] Simule a execução do algoritmo desenvolvido na questão anterior no grafo abaixo. Considere que os vértices são visitados pelo seu algoritmo em ordem lexicográfica, e assuma também que cada uma das listas de adjacências também está ordenada em ordem lexicográfica. Mostre os tempos de descoberta e finalização de cada um dos vértices.



14. ★ [CLRS] Utilizando as mesmas regras para a ordem de visitação dos vértices do exercício anterior, execute o algoritmo de busca em profundidade no grafo abaixo. Considerando que os vértices representam tarefas e um arco (u, v) indica que a tarefa u precisa ser feita antes de v , dê também uma ordenação topológica para as tarefas representadas pelo grafo.



15. ★ Mostre por quê não é possível fazer uma ordenação topológica dos vértices de um grafo que contenha ciclos.
16. ★ Escreva um algoritmo que decide se existe um ciclo em um grafo não orientado dado como entrada. Mudaria algo caso o grafo fosse orientado?
17. ★ Altere o algoritmo de Prim visto em aula para que ele receba um grafo representado por uma matriz de adjacências.
18. ★ [CLRS] Usando o grafo abaixo e tomando como origem o nó a , demonstre passo a passo a execução do algoritmo de Prim visto em aula. Indique a cada passo quais arestas já fazem parte da árvore geradora de custo mínimo.



19. ★★ O Prof. Espertalhão diz ter inventado uma maneira muito eficiente para determinar a árvore geradora de custo mínimo de um grafo. Ele argumenta que para fazê-lo basta pegar todas as arestas, ordená-las pelo seus custos e para cada uma delas, em ordem não decrescente, fazer o seguinte:
 - Comece com uma árvore geradora vazia
 - Enquanto ainda existirem vértices ainda não alcançados pela árvore geradora de custo mínimo repita
 - Retire a aresta de menor peso do conjunto de arestas
 - Se as duas extremidades (vértices) da aresta estiverem no conjunto de vértices já alcançados pela árvore geradora mínima construída até este momento, descarte a aresta
 - Senão, adicione a aresta à árvore geradora sendo construída

O algoritmo proposto pelo Prof. Espertalhão funciona?