



Exercícios de fixação

1. Considere os grafos abaixo e responda os seguintes itens:

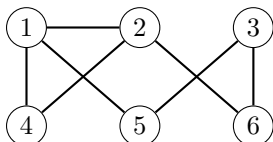


Figura (a) - Grafo não-direcionado

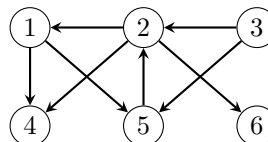


Figura (b) - Grafo direcionado

- Quais vértices são adjacentes ao vértice 2?
 - Na Figura (a), quais vértices possuem grau 2? Na Figura (b), quais são os graus de entrada e de saída do vértice 5?
 - Identifique um caminho do vértice 1 ao vértice 3.
 - Identifique um ciclo em cada um dos grafos.
 - O grafo da Figura (a) é conexo? E para grafos orientados, o que significa ser fortemente conexo?
2. Represente os grafos do exercício anterior por:
- Matriz de adjacência
 - Lista de adjacência
3. Considerando grafos não-orientados, qual o número máximo de arestas em um grafo simples (sem arestas paralelas e auto-laço) com V vértices? Em um grafo com V vértices, qual o número mínimo de arestas necessárias para que este seja conexo?
4. Dada uma representação de lista de adjacências de um grafo orientado, qual o tempo necessário para computar o grau de saída de todos os vértices? Qual o tempo necessário para computar os graus de entrada? [Cormen – 22.1-1]
5. O grafo **transposto** do grafo orientado $G = (V, E)$ é o grafo $G^T = (V, E^T)$, onde $E^T = \{(v, u) \in V \times V : (u, v) \in E\}$. Deste modo, G^T é G com todas as suas arestas invertidas. Descreva algoritmos eficientes para calcular G^T a partir de G , para a representação de lista de adjacências e também para a representação de matriz de adjacências de G . Analise os tempos de execução de seus algoritmos. [Cormen – 22.1-3]
6. Dado um grafo G representado por matriz de adjacência, descreva um algoritmo que devolve o complemento de G . Analise a complexidade do algoritmo que você desenvolveu. Faça o mesmo para o caso em que G é representado por lista de adjacência.
7. Mostre os valores de **dist** e **pred** que resultam da execução da busca em largura sobre o grafo orientado da Figura 22.2(a), usando o vértice 3 como origem. [Cormen – 22.2-1]

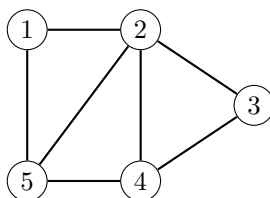


Figura 22.2(a)

8. Existem dois tipos de lutadores profissionais: “bons sujeitos” e “maus sujeitos”. Entre qualquer par de lutadores profissionais pode ou não haver uma rivalidade. Suponha que temos n lutadores profissionais e temos uma lista de r pares de lutadores para os quais existem rivalidades. Dê um algoritmo de tempo $O(n + r)$ que determine se é possível designar alguns dos lutadores como bons sujeitos e os restantes como maus sujeitos, de tal forma que a rivalidade ocorra em cada caso entre um bom sujeito e um mau sujeito. Se for possível realizar tal designação, seu algoritmo deve produzi-la. [Cormen – 22.2.6]