Lista de exercícios 03

Os seguintes exercícios são para aprofundar no estudo da matéria.

Grafos

1. Responda verdadeiro ou falso. Justifique cada caso mediante prova ou contraexemplo:

(a) Um grafo G é uma árvore se e somente se G for acíclico maximal (ou seja, G é acíclico e a adição de qualquer aresta cria um ciclo).

(b) Se em um grafo G todos os vértices tem grau no conjunto $\{1, 2\}$ e há exatamente dois vértices com grau 1, então G é exatamente um caminho.

(c) Um grafo bipartido com n vértices tem no máximo $\frac{n^2}{4}$ arestas.

(d) Se d é o grau mínimo de um grafo G, então em G há um caminho de comprimento maior ou igual que d.

2. Um *k-cubo* é um grafo cujos vértices são todas as sequências $b_1 b_2 \ldots b_k$ em que cada b_i pertence a $\{0, 1\}$. Em um k-cubo, dois vértices são adjacentes se diferem em exatamente uma posição. Quantas arestas tem o k-cubo? Prove que um k-cubo é bipartido.

3. Uma *grade p-por-q* é um grafo cujo conjunto de vértices é o produto cartesiano $\{1, 2\ \ldots, p\} \times \{1, 2, \ldots, q\}$ e dois vértices (i, j) e (i', j') são adjacentes se: $|i - i'| = 1$ e $j = j'$, ou $i = i'$ e $|j - j'| = 1$. Quantas arestas tem a grade p-por-q? Mostre que a grade p-por-q é um grafo bipartido.

4. O *diâmetro* de um grafo G é a maior distância entres dois vértices de G.

(a) Seja G um grafo simples de diâmetro maior ou igual que três. Mostre que $\overline{G}$ tem diâmetro no máximo três.

(b) Seja T uma árvore com pelo menos uma aresta de diâmetro d. Suponha que após a remoção de uma aresta, obtemos duas componentes conexas T_1 e T_2 com diâmetro d_1 e d_2, respectivamente. Mostre que $d \geq (d_1 + d_2)/2 + 1$.

5. Dê um algoritmo que determine se um determinado grafo não direcionado $G = (V, E)$ contém um ciclo. Seu algoritmo deve ser executado em $O(V)$, independente de $|E|$.

6. 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 $O(V + E)$.

7. Um ponto de articulação de um grafo G é um vértice cuja remoção provoca o aumento de componentes conexas. Projete um algoritmo para encontrar pontos de articulação em tempo $O(V + E)$.

8. Projete um algoritmo que recebe um digrafo acíclico G e dois vértices s e t e devolve o número de caminhos diferentes em G de s até t. Seu algoritmo deve executar em tempo $O(V + E)$.

9. Um *ralo universal* em um digrafo $G = (V, E)$ é um sorvedouro com grau de entrada $|V| - 1$ (no qual chega um arco desde cada vértice do grafo). Dada a matriz de adjacências de um digrafo G, projete um algoritmo $O(V)$ para determinar se G possui um ralo universal.

10. Seja T uma árvore produzida pelo *BFS* de um grafo G. Embora os conceitos de arcos de avanço, cruzamento e retrocesso tenham sido definidos no contexto da busca *DFS*, eles fazem sentido em relação a qualquer árvore ou floresta radicada de G.

(a) Mostre que G não tem arcos de avanço em relação a T.

(b) Mostre que, se G for não-dirigido, então G não tem arcos de retorno.

(c) G pode ter arcos de cruzamento em relação a T? E se G for não-dirigido?

Grafos ponderados

11. Suponha que é dado um grafo G com custos nas arestas positivos e diferentes. Seja T uma árvore geradora mínima de G. Agora suponha que substituímos o custo cada aresta, c_e, por seu quadrado, c_e^2, criando assim uma nova instância do problema com o mesmo grafo, mas com custos diferentes. Concorde ou descorde: T ainda deve ser uma árvore geradora mínima para a nova instância. Prove ou dê um contraexemplo.

12. Seja $G = (V, E)$ um grafo conexo com pesos associados a cada aresta. O professor B. Smart propôs o seguinte algoritmo para encontrar uma árvore geradora mínima de G.

```
input   : grafo conexo G = (V, E) e função de pesos nas arestas ω
output  : H subgrafo de G.
begin
    Ordene E em ordem não crescente de pesos
    H ← G
    for e ∈ E em ordem não crescente de pesos do
        if H − e é conexo then
            H ← H − e
        end
    end
    return H
end
```

Algorithm 1: SMART-AGM(G, ω).

(a) Argumente sucintamente (em poucas linhas) que o grafo obtido é conexo e acíclico.

(b) Mostre que se e é uma aresta de G com peso máximo e $G - e$ é conexo, então existe uma árvore geradora mínima de G que não contém e.

(c) Usando os resultados dos itens anteriores, mostre que o algoritmo está correto.

13. Responda as seguintes questões:

(a) Mostre que, se para todo $X \subseteq V$, o corte $\delta(X)$ contém exatamente uma aresta leve, então existe uma única árvore geradora mínima T de G.

(b) Suponha que todas as arestas de G têm custos distintos, com exceção de duas $(u, v), (x, y)$, que têm o mesmo custo. Suponha também que todo caminho de u até x contém essas duas arestas. Argumente que existe uma única árvore geradora mínima. Você pode utilizar o fato enunciado no item anterior.

14. Seja $G = (V, E)$ um grafo não direcionado. Um conjunto $F \subseteq E$ de arestas é chamado conjunto de retroalimentação se cada ciclo de G tiver pelo menos uma aresta em F.

(a) Suponha que G não seja ponderado. Crie um algoritmo eficiente para encontrar um conjunto de retroalimentação de tamanho mínimo.

(b) Suponha que G é um grafo não direcionado com pesos positivos nas arestas. Projete um algoritmo eficiente para encontrar um conjunto de retroalimentação de peso mínimo. Argumente que seu algoritmo está correto.

15. Uma empresa possui n filiais que devem ser conectadas, direta ou indiretamente. Cada par de filiais que são conectadas diretamente utiliza um serviço de fibra ótica, ou um serviço de conexão via linha telefônica. A velocidade da conexão (em MBit/s) via linha telefônica entre duas filiais depende da distância e é fornecida pela empresa de telefonia. A velocidade da fibra é sempre constante, 1GBit/s. Devido a restrições orçamentárias, somente f $(f < n)$ conexões via fibra serão contratadas. Uma vez instalada a rede, duas filiais podem ser conectadas (indiretamente) por uma sequência de conexões diretas; *a velocidade de conexão entre as duas filiais* é a velocidade da conexão mais lenta nessa sequência. A velocidade de rede é a menor velocidade de conexão entre quaisquer duas filiais. Escreva um algoritmo para encontrar uma rede com a maior velocidade de rede possível. Argumente por que o algoritmo está correto e analise a complexidade.

16. Considere um grafo direcionado $G = (V, E)$ cujas arestas têm pesos 0 ou 1. Projete um algoritmo de tempo $O(V + E)$ que obtém uma árvore de caminhos mínimos a partir de um vértice s.

17. Dado um grafo ponderado e direcionado $G = (V, E)$ sem ciclos de peso negativo, seja m o máximo, entre todos os vértices $v \in V$, do número mínimo de arestas em um caminho mínimo da fonte s para v. (Aqui, o caminho mínimo é por peso, e não por número de arestas.) Reescreva o algoritmo Bellman-Ford para ele termine em $m + 1$ passos, mesmo que m não seja conhecido com antecedência.

18. Escreva um algoritmo que verifica se há ciclos negativos em um grafo direcionado e, se houver, devolva um tal ciclo.

19. Milda é a presidenta de um determinado país B. Esse país é dividido em estados e cada estado possui uma capital. Milda quer reestruturar o sistema de estradas e ferrovias e precisa da sua ajuda. As ferrovias são muito antigas e seu custo de manutenção é alto. Seu trabalho é ajudar a presidenta a decidir quais ferrovias podem ser desativadas. No entanto, como B é um país democrático, uma ferrovia só pode ser desativada se isso não piorar a qualidade do sistema de transporte, isso é, uma ferrovia pode ser desativada apenas se a distância de cada cidade à capital mais próxima não for modificada. Considere que o país B tem n cidades e que se uma estrada ou ferrovia liga a cidade i à cidade j, então ela pode ser utilizada para ir tanto de i para j como de j para i. Obtenha um algoritmo eficiente para descobrir quantas ferrovias podem ser desativadas.

20. Dados vértices s e t em um *grafo direcionado* G, dizemos que uma coleção $P_1, P_2, \ldots, P_k$ de caminhos com início em s e final em t é *aresta-disjunta* se para quaisquer caminhos P_i e P_j, P_i e P_j não têm arestas em comum. Demonstre que, dado um grafo G e vértices s e t, o número máximo de caminhos disjuntos nas arestas de s a t é igual ao número mínimo de arestas necessárias para desconectar s de t.

Reduções

21. Sejam P_1 e P_2 dois problemas tais que $P_1 \preccurlyeq_n P_2$ e suponha que P_1 tem cota inferior $\Omega(n \log n)$, onde n é um parâmetro que mede o tamanho da entrada do problema P_1. Quais das seguintes afirmações são verdadeiras? Justifique cuidadosamente as suas respostas.

(a) $\Omega(n \log n)$ também é cota inferior para P_2.

(b) Todo algoritmo que resolve P_1 também pode ser usado para resolver P_2.

(c) Todo algoritmo que resolve P_2 também pode ser usado para resolver P_1.

(d) O problema P_2 pode ser resolvido no pior caso em tempo $O(n \log n)$.

22. Sejam P_1 e P_2 dois problemas tais que um deles tenha cota inferior $\Omega(n^k)$, para algum $k > 1$, e o outro é solúvel em tempo $O(n \log n)$. Se P_1 é redutível a P_2 em tempo linear, diga qual é qual. O parâmetro n denota o tamanho da entrada dos dois problemas.

23. Sejam A e B dois problemas tais que $A \preccurlyeq_{n \log n} B$ e suponha que A tem cota inferior $\Omega(n^{3/2})$, onde n é um parâmetro que mede o tamanho de uma instância I_A de A. Suponha que dada I_A, a redução obtém uma instância I_B de B. Quais das seguintes afirmações são verdadeiras? Justifique.

(a) A notação $A \preccurlyeq B$ significa que o pior algoritmo para A é mais rápido que o melhor algoritmo para B.

(b) Se existe algoritmo $O(n^{3/2})$ para B, então o problema A pode ser resolvido em tempo $O(n \log n)$.

(c) Se a instância I_B tem tamanho $\Theta(n)$, então $\Omega(n^{3/2})$ também é cota inferior para B.

(d) Se não existe um algoritmo que resolve A, então também não existe um algoritmo que resolve B.

(e) Se existe um algoritmo que resolve A, então também existe um algoritmo que resolve B.

24. Responda verdadeiro ou falso. Justifique em cada caso via prova ou contraexemplo.

(a) Se existe problema $L \in \mathsf{NP}$ tal que $L \notin \mathsf{NP}$-completo, então para todo $L' \in \mathsf{NP}$-completo, não existe redução $L' \preccurlyeq_p L$.

(b) Qualquer linguagem pertencente a NP pode ser decidida por um algoritmo em tempo $2^{O(n^k)}$, para alguma constante k.

(c) Se $\mathsf{NP} \neq \mathsf{co\text{-}NP}$, então $\mathsf{P} \neq \mathsf{NP}$.

(d) Se $L_1 \preccurlyeq_p L_2$ e $L_2 \preccurlyeq_p L_3$ então $L_1 \preccurlyeq_p L_3$.

(e) $L \preccurlyeq_p \overline{L}$ se e somente se $\overline{L} \preccurlyeq_p L$.

25. Considere o problema para decidir se um número p é primo. Mostre que esse problema está em co-NP. Atente-se para o tamanho da codificação de um número p.

c

NP-complexidade

26. Dado um conjunto de elementos $E = \{1, 2, ..., n\}$ e uma família de conjuntos $\mathcal{S} = \{S_1, S_2, ..., S_m\}$, onde $S_i \subseteq E$, uma cobertura de conjuntos é uma subfamília $F \subseteq \mathcal{S}$ tal que $\bigcup_{S \in F} S = E$. O problema da cobertura por conjuntos é: dados E, $\mathcal{S}$ e k, existe uma cobertura de conjuntos F de tamanho $|F| \leq k$? Mostre que cobertura por conjuntos é NP-completo.

27. Considere o problema do empacotamento (BIN). Dado um conjunto finito de n objetos com pesos $w_1, w_2, \dots, w_n$ inteiros positivos e dois valores inteiros positivos W e k, é possível colocar todos os objetos em k caixas cujo limite máximo de peso é W? Mostre que este problema é NP-completo.

28. Prove que o problema a seguir é NP-completo: dado um grafo $G = (V, E)$ e um inteiro positivo k, determine se G contém uma árvore geradora T tal que todo vértice em T tenha grau no máximo k.

29. Um ciclo de um grafo é Hamiltoniano se passar por todos os vértices do grafo. Analogamente, um caminho que passa por todos os vértices é Hamiltoniano:

(a) Considerando que é NP-completo o problema de decidir se um grafo tem um caminho Hamiltoniano, mostre que o de decidir se um grafo tem um ciclo Hamiltoniano é também NP-completo.

(b) Considerando que é NP-completo o problema de decidir se um grafo tem um ciclo Hamiltoniano, mostre que o de decidir se um grafo tem um caminho Hamiltoniano é também NP-completo.

30. Uma clique de tamanho é um subgrafo completo de um grafo G, Um conjunto independente é um conjunto de vértices no qual não há dois vértices adjacentes:

(a) Considerando que é NP-completo o problema de decidir se um grafo tem uma clique com pelo menos C vértices, mostre que o de decidir se um grafo tem um conjunto independente com pelo menos I vértices é também NP-completo.

(b) Considerando que é NP-completo o problema de decidir se um grafo tem um conjunto independente com pelo menos I vértices, mostre que o de decidir se um grafo tem uma clique com pelo menos C vértices é também NP-completo.

31. Uma cobertura de vértices de um grafo é um subconjunto S dos vértices no qual toda aresta tem pelo menos um extremo em S. Mostre que o problema de decidir se um grafo tem uma cobertura de vértices de tamanho no máximo k é NP-completo.

32. Um conjunto dominante de um grafo é um subconjunto S dos vértices no qual todo vértice que esteja fora de S tem pelo menos um adjacente em S. Mostre que o problema de decidir se um grafo tem um conjunto dominante de tamanho no máximo k é NP-completo.

33. Dada uma sequência de números positivos $S = \{s_1, s_2, \dots, s_n\}$ e um número $b > 0$, é NP-completo decidir se existe um subconjunto dos índices $X \subseteq \{1, , n\}$ tal que $\sum_{i \in X} s_i = b$. A partir desse problema, mostre que as seguintes variantes também são NP-completos:

(a) Dada uma sequência de números positivos $S = \{s_1, s_2, \dots, s_n\}$, decidir se existe um subconjunto dos índices $X \subseteq \{1, , n\}$ tal que $\sum_{i \in X} s_i = \sum_{i \notin X} s_i$.

(b) Dada uma sequência de números $S = \{s_1, s_2, \dots, s_n\}$, decidir se existe um subconjunto não vazio dos índices $X \subseteq \{1, , n\}$ tal que $\sum_{i \in X} s_i = 0$.