Universidade Estadual de Maringá
Departamento de Informática – Projeto e Análise de Algoritmos
Prof. Daniel Kikuti

Lista de exercícios: Complexidade Computacional

1. Marque (V)erdadeiro ou (F)also.
   (a) ☐ V ☐ F O problema de verificar se um número x pertence a um conjunto de n números está em P .
   (b) ☐ V ☐ F Se P $\neq$ NP então nenhum problema NP pode ser resolvido em tempo polinomial.
   (c) ☐ V ☐ F Se P $\neq$ NP então nenhum problema NP-Completo pode ser resolvido em tempo polinomial.
   (d) ☐ V ☐ F O problema de verificar se uma fórmula booleana é satisfazível pertencente à classe NP.
   (e) ☐ V ☐ F Se há um algoritmo de tempo $O(n^{100})$ para o problema de Subset-Sum, então P = NP.
   (f) ☐ V ☐ F Suponha que $X \in$ NP. Se existir um algoritmo de tempo $O(\lg n)$ para X, então P = NP.
   (g) ☐ V ☐ F O problema de parada da Máquina de Turing é NP-Completo.
   (h) ☐ V ☐ F Se um problema X é NP-Completo, então existe um algoritmo de tempo polinomial não-determinístico que resolve X.

2. Um problema X é NP-Difícil se e somente se existe um problema NP-Completo Y tal que Y $\preceq_p$ X. X não precisa estar em NP e nem ser um problemas de decisão. Seja S um problema NP-Completo, Q e R problemas que não sabemos se pertencem a NP. Se existe uma redução em tempo polinomial de Q para S e, S pode ser reduzido em tempo polinomial para R, assinale qual(is) sentença(s) é(são) verdadeira(s)?
   ☐ R é NP-Completo ☐ R é NP-Difícil ☐ Q é NP-Completo ☐ Q é NP-Difícil

3. O algoritmo de programação dinâmica para o problema da Mochila-Binária é um algoritmo polinomial? Justifique sua resposta (CLRS 34.1-4).

4. Mostre um exemplo em que um número polinomial de chamadas a subrotinas que executam em tempo polinomial pode resultar em um algoritmo de tempo exponencial (CLRS 34.1-5).

5. Considere o problema de determinar se um número natural n é um quadrado perfeito, isto é, se existe um número natural x tal que $x^2 = n$ (Quadrado-Perfeito $= \{\langle n \rangle : \exists x \in \mathbb{N} \mid x^2 = n\}$). É verdade que o problema Quadrado-Perfeito $\in$ P? Justifique. [Dica: Lembre-se que um número n representado na base b ($b > 1$) possui k dígitos, onde $k = \lfloor \log_b n \rfloor + 1$. A complexidade de um algoritmo é dada em função do tamanho da entrada.]

6. Mostre que se existe um problema L pertencente a NP e que não está em P, então nenhum problema NP-Completo está em P.

7. No problema N3DM (*Numerical 3-dimensional matching*) são dados três multiconjuntos de inteiros X, Y e Z, cada um contendo k elementos, e um valor b. O objetivo é determinar se existe um subconjunto M de $X \times Y \times Z$ tal que cada inteiro em X, Y e Z ocorre exatamente uma vez e, para cada tripla (x, y, z) do subconjunto, $x + y + z = b$. Por exemplo, seja $X = \{3, 4, 4\}$, $Y = \{1, 4, 6\}$, $Z = \{1, 2, 5\}$ e $b = 10$. Esta instância possui uma solução: $\{(3, 6, 1), (4, 4, 2), (4, 1, 5)\}$. Note que o subconjunto $\{(3, 6, 1), (3, 4, 2), (4, 1, 5)\}$ não é uma solução, pois viola várias condições (um $4 \in X$ não é usado, o $3 \in X$ é usado mais de uma vez, a segunda tripla não soma 10). Mostre que N3DM $\in$ NP.

8. Um algoritmo verificador para um problema de decisão recebe dois objetos: uma instância do problema e um certificado. Ao receber esses dois objetos, o verificador pode responder SIM ou NÃO. Se responder SIM, dizemos que o verificador aceitou o certificado. Um verificador para um determinado problema de decisão é polinomial se: (i) para cada instância positiva do problema, existe um certificado que o verificador aceita em tempo limitado por uma função polinomial no tamanho da instância; (ii) para cada instância negativa do problema, não existe certificado que o verificador aceite. Dada esta definição, apresente um algoritmo verificador para o Problema de Cobertura de Vértice. [Dado um grafo $G = (V, E)$, uma cobertura de vértices é um subconjunto de vértices $S \subseteq V$ tal que toda aresta $(u, v) \in E$ tem pelo menos uma ponta (u ou v) em S.]

9. Mostre que Vertex-Cover $= \{\langle G, k \rangle : G$ possui uma cobertura de vértices de tamanho $k\}$ é NP-Completo.

10. Um clique em um grafo $G = (V, E)$ é um subgrafo completo de G (cada dois vértices do subconjunto são conectados por uma aresta). Encontrar um clique de tamanho máximo em um grafo é um problema de otimização para o qual não se conhece um algoritmo que o resolva em tempo polinomial. O problema de decisão que verifica se um grafo possui um clique de tamanho $\geq k$ é NP-Completo. Mas se P=NP, então mostre que:
    (a) Existe um algoritmo que resolve o Problema do Clique Máximo em tempo polinomial (devolve o tamanho do Clique Máximo). [Note que, Assuma que Clique(G, k) é o algoritmo que resolve este problema. Dica: desenvolva seu algoritmo de modo a fazer diversas consultas a Clique(G, k).]

(b) Considerando a solução do item anterior, apresente um algoritmo que devolve em tempo polinomial os vértices que fazem parte do Clique Máximo.

11. Um conjunto independente de um grafo $G = (V, E)$ é um conjunto de vértices $S \subseteq V$ tal que não existem dois vértices adjacentes contidos em S.

    (a) Mostre que o problema de encontrar um conjunto independente de tamanho k em um grafo é um problema NP-Completo.

    (b) E o problema de encontrar em um grafo um conjunto independente contendo exatamente 3 vértices? Também é um problema NP-Completo? Justifique.

12. Mostre que Solve $\in$ NP. No problema Solve são dados uma tupla de inteiros $A[1..n]$ e um valor k. O objetivo é determinar se existe uma expressão aritmética

$$A[1] \odot_1 A[2] \odot_2 \ldots \odot_{n-1} A[n] = k,$$

    tal que $\odot_i$ $(1 \leq i \leq n-1)$ substituído por algum dos operadores elementares $\{+, -\}$ resulta no valor k. Por exemplo, as instâncias $\langle\{1,1,1,1,1\};10\rangle$ e $\langle\{1,1,1,1,1\};0\rangle$ resultam em rejeição; as instâncias $\langle\{1,2,3,4\};2\rangle$ e $\langle\{2,3,5\};4\rangle$ resultam em aceitação.

13. Seu amigo apresentou a seguinte “prova” de que P $\neq$ NP: “Para verificar se uma fórmula booleana com n variáveis é satisfazível, podemos construir uma tabela verdade e verificar ao todo 2^n atribuições de valores para as variáveis. Isto leva tempo exponencial, portanto, o problema de satisfazibilidade de fórmulas booleanas (Sat) não pertence a P. Como Sat está em NP, logo, podemos concluir que P $\neq$ NP.” Explique qual o equívoco na demonstração de seu amigo.

14. Dados os problemas:

    - Subset-Sum $= \left\{\langle S, t\rangle : \text{existe } S' \subseteq S \text{ tal que } t = \sum_{s \in S'} s\right\}$
    - Busca-Inteiro $= \{\langle A, r\rangle : r \in A\}$

    Considere a seguinte redução de Subset-Sum para Busca-Inteiro: (1) enumere todos os subconjuntos de S e compute a soma dos elementos de cada um destes subconjuntos; (2) a soma dos elementos destes subconjuntos são elementos do conjunto A; (3) seja $r = t$. Por exemplo, para $S = \{1,2,3\}$ e $t = 5$, teríamos $A = \{1,2,3,3,4,5,6\}$ e $r = 5$. Note que existe um subconjunto S' que soma t se e somente se o conjunto A possui o elemento r.

    Como Subset-Sum $\in$ NP-Completo e Busca-Inteiro $\in$ P, pela redução acima é possível concluir que P = NP? Justifique.

15. Vimos um algoritmo de Programação Dinâmica que resolve uma variante do problema de soma de subconjuntos Subset-Sum $= \left\{\langle S, t\rangle : \text{existe } S' \subseteq S \text{ tal que } t = \sum_{s \in S'} s\right\}$ em tempo $O(|S| \cdot t)$, dado que t e todos os elementos em S são positivos. Mostre que a versão em que S pode conter números negativos não é mais difícil que a versao vista em aula, isto é, que há uma redução em tempo polinomial desta versão com números negativos para aquela vista em aula.

16. Um quadrado latino (*latin square*) de ordem n é uma matriz $n \times n$ preenchida com n diferentes símbolos de tal maneira que estes ocorrem no máximo uma vez em cada linha ou coluna. A Figura (a) apresenta um exemplo de quadrado latino. Dada uma matriz $n \times n$ parcialmente preenchida, é possível formar um quadrado latino? As Figuras (b) e (c) apresentam configurações em que isto é possível e não é possível respectivamente.

A	B	C
B	C	A
C	A	B

(a)

A	B	
B	C	
C		B

(b)

A	B	
B	A	
C		B

(c)

    Considerando o problema Latin-Square $= \{\langle M_{n\times n}, \Sigma\rangle \mid M$ é uma matriz parcialmente preenchida com símbolos do alfabeto Σ em que é possível formar um quadrado latino$\}$. Mostre que Latin-Square $\in$ NP.

17. Mostre que 4-Sat $\in$ NP-Completo. [Problema de satisfazibilidade de fórmula booleana na qual a fórmula está na forma normal conjuntiva e possui 4 literais por cláusula.]

18. Mostre que 4-Sat $\preceq_p$ 3-Sat. Se sabemos que 3-Sat é NP-Completo, podemos afirmar que 4-Sat também é NP-Completo por meio desta redução? Justifique.

19. Sabe-se que 3-SAT é NP-Completo e que existe um algoritmo de tempo polinomial para o problema 2-SAT. Se for provado que P $\neq$ NP, então é possível termos 3-SAT $\preceq_p$ 2-SAT? Justifique. [Dica: a questão envolve diversos conceitos (NP-Completo, P $\neq$ NP e redução em tempo polinomial) que precisam estar bem definidos para se ter uma justificativa consistente. Certifique-se de que sua resposta explica o que são estes conceitos e deixa claro o porquê da possibilidade ou impossibilidade da redução.]

20. O problema da mochila binária é um problema de otimização cujo objetivo é escolher itens para se colocar na mochila, de modo a maximizar a soma dos valores dos itens, não ultrapassando a capacidade de peso da mochila. A versão de decisão equivalente deste problema consiste em saber se existe uma subcoleção dos itens que não excede o peso W e cuja soma dos valores é maior ou igual a k, isto é:

$$\text{MOCHILA-DEC} = \{\langle A, W, k\rangle : \exists S \subseteq A | \sum_{s \in S} w_s \leq W, \sum_{s\in S} v_s \geq k\}.$$

    Suponha que alguém demonstrou que MOCHILA-DEC $\in P$. Assuma que todos os valores dos itens são inteiros. Descreva um algoritmo que faz uso de MOCHILA-DEC e resolve o problema de otimização. Podemos afirmar que o problema de otimização também pertence à P? Justifique.

21. Mostre que MOCHILA-DEC é NP-Completo.

22. O problema de partição de conjunto recebe como entrada um conjunto S de números. A questão é determinar se os números podem ser particionados em dois subconjuntos A e $\overline{A} = S \setminus A$ tal que $\sum_{x\in A} x = \sum_{x \in \overline{A}} x$. Mostre que SET-PARTITION é NP-Completo.

23. Defina a classe de problemas co-NP. Mostre um exemplo de problema que pertence a esta classe.

24. É verdade que se NP $\neq$ co-NP, então P $\neq$ NP? Justifique.