Introdução à Teoria da Computação Exercícios

Livro: Michel Sipser, Introdução à Teoria da Computação – 2ª Ed. – Capítulo 07

EXERCÍCIOS

7.1 Responda VERDADEIRO ou FALSO para cada parte.

a. $2n = O(n)$.

b. $n^2 = O(n)$.

^{R}**c.** $n^2 = O(n \log^2 n)$.

^{R}**d.** $n \log n = O(n^2)$.

e. $3^n = 2^{O(n)}$.

f. $2^{2^n} = O(2^{2^n})$.

7.2 Responda VERDADEIRO ou FALSO para cada parte.

a. $n = o(2n)$.

b. $2n = o(n^2)$.

^{R}**c.** $2^n = o(3^n)$.

^{R}**d.** $1 = o(n)$.

e. $n = o(\log n)$.

f. $1 = o(1/n)$.

7.3 Quais dos seguintes pares de números são primos entre si? Mostre os cálculos que levaram às suas conclusões.

a. 1274 e 10505

b. 7289 e 8029

7.4 Preencha a tabela descrita no algoritmo de tempo polinomial para reconhecimento de linguagem livre-do-contexto do Teorema 7.16 para a cadeia $w = \texttt{baba}$ e a GLC G:

$$\begin{aligned} S &\to RT \\ R &\to TR \mid \texttt{a} \\ T &\to TR \mid \texttt{b} \end{aligned}$$

7.5 A fórmula a seguir é satisfazível?

$$(x \vee y) \wedge (x \vee \overline{y}) \wedge (\overline{x} \vee y) \wedge (\overline{x} \vee \overline{y})$$

7.6 Mostre que P é fechada sob união, concatenação e complementação.

7.7 Mostre que NP é fechada sob união e concatenação.

7.8 Seja $CONEXO = \{\langle G\rangle \mid G$ é um grafo não-direcionado conexo$\}$. Analise o algoritmo dado na página 165 para mostrar que essa linguagem está em P.

7.9 Um ***triângulo*** em um grafo não-direcionado é um 3-clique. Mostre que $TRI\hat{A}NGULO \in$ P, onde $TRI\hat{A}NGULO = \{\langle G\rangle \mid G$ contém um triângulo$\}$.

7.10 Mostre que $TODAS_{\mathsf{AFD}}$ está em P.

7.11 Chame os grafos G e H ***isomorfos*** se os nós de G podem ser reordenados de modo que ele fique idêntico a H. Seja $ISO = \{\langle G, H\rangle \mid G$ e H são grafos isomorfos$\}$. Mostre que $ISO \in$ NP.

PROBLEMAS

7.12 Seja

$$EXPMOD = \{\langle a, b, c, p\rangle \mid a, b, c \text{ e } p \text{ são inteiros em binário tais que } a^b \equiv c \pmod{p}\}.$$

Mostre que $EXPMOD \in$ P. (Note que o algoritmo mais óbvio não roda em tempo polinomial. Dica: Tente primeiro com b sendo uma potência de 2.)

7.13 Uma ***permutação*** sobre o conjunto $\{1, \ldots, k\}$ é uma função bijetora sobre o mesmo. Quando p é uma permutação, p^t denota a composição de p com si mesma t vezes. Seja

$$POT\text{-}PERM = \{\langle p, q, t\rangle \mid p = q^t \text{ onde } p \text{ e } q \text{ são permutações sobre } \{1, \ldots, k\} \text{ e } t \text{ é um inteiro em binário}\}.$$

Mostre que $POT\text{-}PERM \in$ P. (Note que o algoritmo mais óbvio não roda em tempo polinomial. Dica: Tente primeiro com t sendo uma potência de 2.)

7.14 Mostre que P é fechada sob a operação estrela. (Dica: Use programação dinâmica. Sobre uma entrada $y = y_1 \cdots y_n$ for $y_i \in \Sigma$, construa uma tabela que indique, para cada $i \leq j$, se a subcadeia $y_i \cdots y_j \in A^*$ para qualquer $A \in$ P.)

R7.15 Mostre que NP é fechada sob a operação estrela.

7.16 Seja *SOMA-SUBCU* o problema da soma de subconjuntos no qual todos os números são representados em unário. Por que a prova de NP-completude para *SOMA-SUBC* falha em mostrar que *SOMA-SUBCU* é NP-completa? Mostre que *SOMA-SUBCU* $\in$ P.

7.17 Mostre que, se P $=$ NP, então toda linguagem $A \in$ P, exceto $A = \emptyset$ e $A = \Sigma^*$, é NP-completa.

***7.18** Mostre que *PRIMOS* $= \{m \mid m$ é um número primo em binário$\} \in$ NP. (Dica: Para $p > 1$ o grupo multiplicativo $Z_p^* = \{x \mid x$ e p são primos entre si e $1 \leq x < p\}$ é, ambos, cíclico e de ordem $p - 1$ sse p é primo. Você pode usar esse fato sem justicá-lo. A afirmativa mais forte *PRIMOS* $\in$ P é atualmente conhecida como verdadeira, mas é mais difícil de provar.)

7.19 Em geral, acredita-se que *CAM* não é NP-completa. Explique a razão por trás dessa crença. Mostre que provar que *CAM* não é NP-completa provaria que P $\neq$ NP.

7.20 Suponha que G represente um grafo não-direcionado. Seja também

$$CAMMÁX = \{\langle G, a, b, k\rangle \mid G \text{ contém um caminho simples de comprimento no máximo } k \text{ de } a \text{ para } b\},$$

e

$$CAMMIN = \{\langle G, a, b, k\rangle \mid G \text{ contém um caminho simples de comprimento no mínimo } k \text{ de } a \text{ para } b\}.$$

a. Mostre que *CAMMÁX* $\in$ P.

b. Mostre que *CAMMIN* é NP-completa. Pode assumir a NP-completude de *CAMHAMN*, o problema do caminho hamiltoniano para grafos não-direcionados.

7.21 Seja *DUPLO-SAT* $= \{\langle\phi\rangle \mid \phi$ tem pelo menos duas atribuições que a satisfazem$\}$. Mostre que *DUPLO-SAT* é NP-completa.

R7.22 Seja *MEIO-CLIQUE* $= \{\langle G\rangle \mid G$ é um grafo não-direcionado que tem um subgrafo completo com pelo menos $m/2$ nós, onde m é o número de nós em $G\}$. Mostre que *MEIO-CLIQUE* é NP-completa.

7.23 Seja $FNC_k = \{\langle\phi\rangle \mid \phi$ é uma fnc-fórmula satisfazível em que cada variável ocorre em, no máximo, k posições$\}$.

a. Mostre que $FNC_2 \in$ P.

b. Mostre que FNC_3 é NP-completa.

7.24 Seja ϕ uma 3fnc-fórmula. Uma ***$\neq$-atribuição*** às variáveis de ϕ é aquela em que cada cláusula contém dois literais com diferentes valores-verdade. Em outras palavras, uma $\neq$-atribuição satisfaz ϕ sem atribuir verdadeiro a três literais em qualquer cláusula.

a. Mostre que a negação de qualquer $\neq$-atribuição a ϕ é também uma $\neq$-atribuição.

b. Seja $\neq$*SAT* a coleção de 3fnc-fórmulas que têm uma $\neq$-atribuição. Mostre que obtemos uma redução em tempo polinomial de *3SAT* para $\neq$*SAT* substituindo cada cláusula c_i

$$(y_1 \vee y_2 \vee y_3)$$

SOLUÇÕES SELECIONADAS

7.1 **(c)** FALSO; **(d)** VERDADEIRO.

7.2 **(c)** VERDADEIRO; **(d)** VERDADEIRO.

7.15 Seja $A \in$ NP. Construa a MTN M para decidir A em tempo polinomial não-determinístico.

$M =$ "Sobre a entrada w:

1. Não-deterministicamente divida w em pedaços $w = x_1 x_2 \cdots x_k$.
2. Para cada x_i, adivinhe não-deterministicamente os certificados que mostrem que $x_i \in A$.
3. Verifique todos os certificados se possível e, então, *aceite*. Caso contrário, se a verificação falhar, *rejeite*."

7.22 Vamos dar uma redução por mapeamento polinomial de *CLIQUE* para *MEIO-CLIQUE*. A entrada para a redução é um par $\langle G, k\rangle$ e a redução produz o grafo $\langle H\rangle$ como saída, onde H é como segue. Se G tem m nós e $k = m/2$, então $H = G$. Se $k < m/2$, então H é o grafo obtido de G pelo acréscimo de j nós, cada um conectado a todos os nós originais e aos outros $j-1$, onde $j = m - 2k$. Assim, H tem $m + j = 2m - 2k$ nós. Observe que G tem um k-clique sse H tem um clique de tamanho $k + j = m - k$ e, portanto, $\langle G, k\rangle \in$ *CLIQUE* sse $\langle H\rangle \in$ *MEIO-CLIQUE*. Se $k > 2m$, então H é o grafo obtido pela adição de j nós a G sem quaisquer arestas adicionais, onde $j = 2k - m$. Assim, H tem $m + j = 2k$ nós e, logo, G tem um k-clique sse H tem um clique de tamanho k. Portanto, $\langle G, k_k\rangle \in$ *CLIQUE* sse $\langle H\rangle \in$ *MEIO-CLIQUE*. Precisamos mostrar também que *MEIO-CLIQUE* $\in$ NP. O certificado é simplesmente o clique.