MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2013

Lista 7

1. Calcule a função prefixo π para o padrão *ababbabbababbababbabb* com o alfabeto $\Sigma = \{a, b\}$.

2. Mostre que se todos os caracteres do padrão $P[1\,..\,m]$ são distintos, o algoritmo ingênuo que busca P em um texto $T[1\,..\,n]$ pode ser modificado para consumir tempo $\mathrm{O}(n)$.

3. Suponha que o padrão P e o texto T são cadeias de caracteres de comprimentos m e n respectivamente, escolhidas aleatoriamente de um alfabeto $\Sigma_d = \{0, 1, \ldots, d-1\}$, onde $d \geq 2$. Mostre que o número esperado de comparações caractere a caractere feita pelo laço implícito na comparação dentro do laço principal do algoritmo ingênuo é

$$(n-m+1)\frac{1-d^{-m}}{1-d^{-1}} \;\leq\; 2(n-m+1).$$

   (Suponha que o algoritmo ingênuo pára as comparações assim que encontra um caractere do padrão que não casa com o correspondente do texto, ou quando encontra uma ocorrência do padrão.) Assim sendo, para cadeias de caracteres aleatórias, o algoritmo ingênuo é bastante eficiente.

4. Suponha que o padrão P pode conter ocorrências de um caracter *vão* $\diamondsuit$, que pode ser casado a uma cadeia arbitrária de caracteres (inclusive com a cadeia vazia). Por exemplo, o padrão $ab\diamondsuit ba\diamondsuit c$ ocorre no texto *cabccbacbacab* de duas maneiras diferentes: *c***ab***cc***ba***cba***c***ab* e *c***ab***ccbac***ba***c***ab*. Note que o caracter vão pode ocorrer um número arbitrário de vezes no padrão, mas não ocorre no texto nenhuma vez. Descreva um algoritmo polinomial para determinar se um tal padrão ocorre em um texto T, e analise o consumo de tempo do seu algoritmo.

5. Mostre como determinar as ocorrências de um padrão $P[1\,..\,m]$ em um texto $T[1\,..\,n]$ examinando a função prefixo para a cadeia de caracteres PT (concatenação de P com T).

6. Descreva um algoritmo linear para determinar se um texto T é uma rotação cíclica de uma outra cadeia de caracteres T'. Por exemplo, *arco* e *coar* são rotações uma da outra.

7. Defina *algoritmo eficiente*. Defina *problema de decisão*. Defina *verificador polinomial para* SIM. Defina *verificador polinomial para* NÃO. Defina as classes P, NP e coNP. Dê um exemplo de um problema em cada uma dessas classes, justificando a sua pertinência à classe.

8. Mostre que SAT está em NP. (Essa é a parte fácil do teorema de Cook.)

9. Uma coleção $\mathcal{C}$ de cláusulas sobre um conjunto X de variáveis booleanas é uma *tautologia* se toda atribuição a X satisfaz $\mathcal{C}$. O problema TAUTOLOGIA consiste em, dado X e $\mathcal{C}$, decidir se $\mathcal{C}$ é ou não uma tautologia. O problema TAUTOLOGIA está em NP? Está em coNP? Justifique suas respostas.

10. O problema 2-SAT consiste na restrição de SAT a instâncias X e $\mathcal{C}$ em que toda cláusula de $\mathcal{C}$ tem exatamente dois literais. Mostre que o 2-SAT está em P, ou seja, descreva um algoritmo polinomial que resolva o 2-SAT.

11. Mostre que 2-COLORAÇÃO está em P.

12. Seja $G = (V, E)$ um grafo. Um conjunto $S \subseteq V$ é *independente* se quaisquer dois vértices de S não são adjacentes. Ou seja, não há nenhuma aresta do grafo com as duas pontas em S. O problema IS consiste no seguinte: dado um grafo G e um inteiro $k \geq 0$, existe um conjunto independente em G com k vértices? Mostre que IS é NP-completo.

13. Seja $G = (V, E)$ um grafo. Uma *3-coloração* de G é uma função $c : V \to \{1, 2, 3\}$ tal que $c(u) \neq c(v)$, para toda aresta $uv \in E$.

    Considere o

    > **Problema** 3-COLORAÇÃO**:** Dado um grafo, determinar se ele tem ou não uma 3-coloração.

    Mostre que o 3-COLORAÇÃO está em NP.

14. Mostre que o problema abaixo é NP-completo.

    > **Problema** PARTIÇÃO**:** Dada uma coleção S de números, decidir se existe uma subcoleção S' de S cuja soma é igual a soma dos números em $S \setminus S'$, ou seja,
    >
    > $$\sum_{x \in S} x = \sum_{x \notin S} x.$$

15. Mostre que o problema abaixo é NP-completo.

    > **Problema** MOCHILA**:** Dado um número W, um número V, um número inteiro positivo n, uma coleção de números $w_1, \ldots, w_n$, e uma coleção de números $v_1, \ldots, v_n$, decidir se existe um subconjunto S de $\{1, \ldots, n\}$ tal que
    >
    > $$\sum_{i \in S} w_i \leq W \quad \text{e} \quad \sum_{i \in S} v_i \geq V.$$