MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2008

Lista 8

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

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 $O(n)$.

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***bac***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.

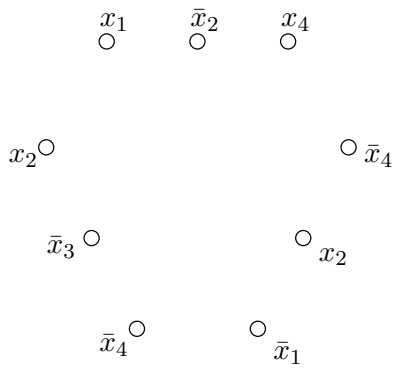
10. O problema 2-SAT consiste na restrição de SAT a instâncias X e $\mathcal{C}$ em que toda cláusula de $\mathcal{C}$ tem exatamente dois literais. Mostre que o 2-SAT está em P, ou seja, descreva um algoritmo polinomial que resolva o 2-SAT.

11. Mostre que 2-COLORAÇÃO está em P.

12. Seja $G = (V, E)$ um grafo. Um conjunto $S \subseteq V$ é um *clique* se existe uma aresta entre quaisquer dois vértices em S. O problema CLIQUE consiste no seguinte: dado um grafo G e um inteiro $k \geq 0$, existe um clique em G com k vértices? Mostre que CLIQUE está em NP.

13. Agora vou ajudar você a construir uma redução do 3-SAT para o CLIQUE. Isso e mais o exercício anterior permitem concluir que CLIQUE é NP-completo. Essa redução aparece em vários livros, mas eu gostaria que você não olhasse a resolução dos livros. Que tentasse fazê-la usando o que segue.

    Para reduzir o 3-SAT para o CLIQUE, lembre-se, você deve considerar uma instância arbitrária do 3-SAT, digamos, X e $\mathcal{C}$, e construir, a partir dela, uma instância do CLIQUE, ou seja, um grafo G e um k. Esses dois últimos devem ser tais que $\mathcal{C}$ é satisfatível se e somente se G tem um clique com k vértices. Com fazer isso? Para cada cláusula em $\mathcal{C}$, o grafo G terá três vértices, cada um rotulado por um dos literais da cláusula. Não há arestas entre estes vértices. Assim, por enquanto, o grafo correspondente a instância $X = \{x_1, x_2, x_3, x_4\}$ e $\mathcal{C} = (x_1 \vee \bar{x}_2 \vee x_4)(x_2 \vee \bar{x}_3 \vee \bar{x}_4)(\bar{x}_1 \vee x_2 \vee \bar{x}_4)$ seria acrescido de



    mais algumas arestas entre as triplas de vértices correspondentes às cláusulas. Denotemos por m o número de cláusulas em $\mathcal{C}$. A sua missão nesse exercício é decidir como ligar os vértices de maneira a obter um grafo que tem um clique com m vértices se e somente se $\mathcal{C}$ é satisfatível. Lembre-se: só vale colocar arestas ligando vértices de "triplas" diferente! (Ou seja, associados a cláusulas diferentes.) Como será um clique de tamanho m se o grafo for obtido assim? Tal clique deve indicar uma atribuição que satisfaça a cláusula... Como devo por as arestas para que só exista um tal clique caso haja uma atribuição que satisfaça todas as cláusulas?

14. 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.