MAC 6711 - Tópicos de Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2010

Lista 6

1. Dados $n + 1$ pontos $p, p_1, \ldots, p_n$, dizemos que p é uma *combinação convexa* de $p_1, \ldots, p_n$ se existem números reais não-negativos $\lambda_1, \ldots, \lambda_n$ tais que

   (a) $\sum_{i=1}^{n} \lambda_i = 1$ e

   (b) $\sum_{i=1}^{n} \lambda_i \, p_i = p$.

   O *fecho convexo* de uma coleção finita de pontos é o conjunto de todas as combinações convexas de pontos da coleção. O *casco convexo* de uma coleção finita de pontos é o polígono que delimita o fecho convexo dessa coleção. Como o casco convexo é um polígono, ele pode ser dado por uma seqüência de pontos: os vértices do polígono. Note que os pontos dessa seqüência são sempre pontos da coleção original de pontos.

   O seguinte algoritmo, proposto por Graham, determina o casco convexo da coleção $\{p_1, \ldots, p_n\}$. Vamos supor, para simplificar, que não há na coleção três pontos colineares. No pseudo-código abaixo, para três pontos distintos p, q e w, denotamos por $\theta(p, q, w)$ o ângulo entre a reta que passa por p e q e a reta que passa por q e w.

   Graham(p, n)
   1 Preliminares (p, n)
   2 $p[n+1] \leftarrow p[1]$
   3 $c[1] \leftarrow p[1]$
   4 $t \leftarrow 1$
   5 **para** $k \leftarrow 2$ **até** n **faça**
   6     $t \leftarrow t + 1$
   7     $c[t] \leftarrow p[k]$
   8   **enquanto** $t > 2$ **e** $\theta(c[t-1], c[t], p[k+1]) \leq 180^o$ **faça**
   9         $t \leftarrow t - 1$
   10 **devolva** c

   Preliminares(p, n)
   1 $min \leftarrow 1$
   2 **para** $i \leftarrow 2$ **até** n **faça**
   3   **se** $y(p[i]) < y(p[min])$
   4   **então** $min \leftarrow i$
   5 $p[1] \leftrightarrow p[min]$
   6 seja q um ponto tal que a reta que passa por q e $p[1]$ é paralela ao eixo x
   7 ordene $p[2\,..\,n]$ de modo que $\theta(q, p[1], p[2]) < \cdots < \theta(q, p[1], p[n])$

   Demonstre que as linhas 2 a 10 do algoritmo de Graham consomem tempo $O(n)$.

2. Considere uma pilha implementada em um vetor com as operações usuais de `empilha` e `desempilha`, usando a técnica de tabelas dinâmicas descrita na aula. Inicialmente a pilha começa vazia. Se, no início da chamada de um `empilha`, for constatado que a pilha não tem mais espaço para o novo elemento, uma nova pilha é alocada com o dobro do tamanho, o conteúdo da velha pilha é copiado para a nova, e a velha é desalocada. Só então o novo elemento é empilhado. Já ao ocorrer um `desempilha`, se a pilha ficar com menos do que um quarto de ocupação, também um novo vetor, com metade do tamanho da pilha corrente é alocado, o conteúdo da velha pilha é copiado para a nova, e a velha é desalocada. Qual é o custo no pior caso de cada `empilha` e `desempilha`? Faça uma análise amortizada e mostre que o custo amortizado de cada `empilha` e `desempilha` é constante. Faça a análise usando os três métodos vistos em aula: o método agregado, o método de créditos e o método do potencial.

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

4. 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 $O(n)$.

5. 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.

6. 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.

7. 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).

8. 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.