MAC5811 Projeto e Análise de Algoritmos

DCC–IME–USP, 2 de março de 2007

Instruções

(*i*) Esta prova contém oito questões sendo seis de dois pontos e duas de três pontos.

(*ii*) A banca considerará questões cujos valores somem até 10 pontos de modo que a soma total das notas obtidas seja máxima. Um aluno, para ser aprovado, precisa obter nessas questões pelo menos 7 pontos.

(*iii*) Enuncie os teoremas e propriedades usados para justificar suas afirmações.

(*iv*) Você pode utilizar como subrotina qualquer algoritmo do CLRS sem reescrevê-lo, como, por exemplo, algoritmos para ordenação. No entanto, você deve descrever clara e sucintamente o que o algoritmo recebe, devolve ou faz e o seu consumo de tempo. Exemplo

> *"O algoritmo* BLÁ-BLÁ-BLÁ *usa como subrotina o algoritmo* ORDENAÇÃO-LERDA (A, n) *que recebe e rearranja um vetor* $A[1 \mathinner{.\,.} n]$ *de modo que ele fique em ordem crescente. O consumo de tempo do algoritmo* ORDENAÇÃO-LERDA *é* $\mathrm{O}(n^n)$.*"*

(*v*) Não é permitida a consulta a livros, anotações, colegas, calculadoras, Internet, computadores ...

Duração da prova: 5 horas

Questão 1 [2 pontos]
Suponha que $f(n)$ e $g(n)$ são funções dos inteiros não-negativos nos inteiros não-negativos. Demonstre que as seguintes afirmações são equivalentes:

- existem números reais $a > 1, b > 1, c > 1$ e existe um número inteiro $n_0 > 0$ tais que $a^{f(n)} \leq b^{g(n)} \leq c^{f(n)}$, para todo $n \geq n_0$;
- $g(n)$ é $\Theta(f(n))$.

Questão 2 [2 pontos]
Considere os problemas a seguir.

Problema Partição(A, n): Dado um vetor $A[1\,..\,n]$ de números inteiros positivos, decidir se existe um subconjunto I de $\{1, \ldots, n\}$ tal que

$$\sum_{i \in I} A[i] = \sum_{i \notin I} A[i] \ .$$

Problema Tri-Partição(A, n): Dado um vetor $A[1\,..\,n]$ de números inteiros positivos, decidir se existem subconjuntos disjuntos I e J de $\{1, \ldots, n\}$ tais que

$$\sum_{i \in I} A[i] = \sum_{i \in J} A[i] = \sum_{i \notin I \cup J} A[i] \ .$$

Sabe-se que o problema Partição é NP-completo. Um aluno alega que o problema Tri-Partição também é NP-completo. O aluno está certo? Justifique cuidadosamente a sua resposta.

Questão 3 [2 pontos]
Suponha que, para entradas de tamanho n, você tenha que escolher um dentre três algoritmos A, B e C.

a) Algoritmo A resolve problemas dividindo-os em **cinco** subproblemas de metade do tamanho, recursivamente resolve cada subproblema e então combina as soluções em tempo $O(n)$.
b) Algoritmo B resolve problemas dividindo-os em **dois** subproblemas de tamanho $n - 1$, recursivamente resolve cada subproblema e então combina as soluções em tempo $O(1)$.
c) Algoritmo C resolve problemas dividindo-os em **nove** subproblemas de tamanho $n/3$, recursivamente resolve cada subproblema e então combina as soluções em tempo $O(n^2)$.

Qual o consumo de tempo de cada um desses algoritmos? Expresse as suas respostas em termos da notação O, mas procure dar as respostas mais justas possíveis. Qual algoritmo é assintoticamente mais eficiente no pior caso? Justifique as suas respostas.

Questão 4 [2 pontos]
Considere o seguinte algoritmo que recebe como entrada um vetor $A[0\,..\,n-1]$ de números inteiros:

Caixa-Preta (A, n)
$k \leftarrow 0 \quad i \leftarrow 1 \quad j \leftarrow 0$
enquanto $i < n$ **e** $k + j + 1 < n$ **faça**
  se $A[k+j] = A[(i+j) \bmod n]$
    então $j \leftarrow j + 1$
  senão se $A[k+j] < A[(i+j) \bmod n]$
    então $i \leftarrow i + j + 1 \quad j \leftarrow 0$
  senão se $A[k+j] > A[(i+j) \bmod n]$
    então $h \leftarrow \max\{i, k+j+1\} \quad k \leftarrow h \quad i \leftarrow h + 1 \quad j \leftarrow 0$
devolva k

Qual o consumo de tempo desse algoritmo? Expresse a sua resposta em termos da notação O, mas procure dar a resposta mais justa possível. Justifique a sua resposta.

Questão 5 [2 pontos]
Considere uma seqüência de n operações

$$\underbrace{\mathrm{O}_1 \quad \mathrm{O}_2 \quad \mathrm{O}_3 \quad \mathrm{O}_4 \quad \mathrm{O}_5 \quad \ldots \quad \mathrm{O}_{n-4} \quad \mathrm{O}_{n-3} \quad \mathrm{O}_{n-2} \quad \mathrm{O}_{n-1} \quad \mathrm{O}_n}_{n \text{ operações Vermelha e Azul, } \mathrm{O}_1 = \text{Vermelha e } \mathrm{O}_2 = \text{Azul}}$$

Cada operação O_i é Vermelha ou Azul sendo que O_1 é Vermelha e O_2 é Azul. O tempo de execução de cada operação Azul é constante. O tempo de execução da primeira operação Vermelha é constante, mas cada operação Vermelha a seguir consome o dobro do tempo da operação Vermelha anterior. Qual o consumo de tempo de uma seqüência de operações Vermelha e Azul em cada uma das situações a seguir. Expresse as suas respostas em termos da notação O, mas procure dar as respostas mais justas possíveis. Justifique as suas respostas.

a) A seqüência possui $\Theta(1)$ operações Azul entre operações Vermelha consecutivas.

b) A seqüência possui $\Theta(\sqrt{n})$ operações Azul entre operações Vermelha consecutivas.

c) Na seqüência, se O_i,O_j e O_k são operações Vermelha consecutivas, $i < j < k$, então o número de operações Azul entre O_j e O_k é pelo menos duas vezes o número de operações Azul entre O_i e O_j.

Questão 6 [2 pontos]
Considere uma estrutura de dados padrão de dicionário com operações Inserir, Buscar e Remover. Agora desejamos ampliar essas operações com Min-Gap que devolve a menor diferença entre os dois números mais próximos no dicionário. Explique como implementar esse dicionário ampliado de tal forma que o consumo de tempo de cada uma das operações Inserir, Buscar, Remover e Min-Gap seja $O(\log n)$ no pior caso, onde n é o número de elementos no dicionário. Você pode utilizar qualquer árvore balanceada de busca como caixa preta.

Questão 7 [3 pontos]
Alice deseja fazer uma festa e está decidindo a quem convidar. Há n pessoas que são candidatas a serem convidadas. Ela preparou uma lista de pares de pessoas que se conhecem. Alice deseja convidar o maior número possível de pessoas de tal forma que na festa cada pessoa conheça pelo menos cinco pessoas e não conheça pelo menos cinco pessoas.

Descreva um algoritmo eficiente que receba como entrada uma lista de n pessoas e uma lista dos pares que se conhecem e devolva uma melhor escolha de pessoas a serem convidadas. Qual o consumo de tempo do seu algoritmo? Uma resposta plenamente satisfatória deve ser um algoritmo que consome tempo $o(n^3)$. O seu algoritmo é plenamente satisfatório? Justifique as suas respostas.

Questão 8 [3 pontos]
Uma seqüência $X[1\,..\,m]$ é subseqüência de uma seqüência $Y[1\,..\,n]$ se existem índices $i_1 < i_2 < \ldots < i_m$ tais que

$$X[j] = Y[i_j] \quad \text{para } j = 1, \ldots, m.$$

Por exemplo,

$$X = \begin{array}{|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} \\ \hline \texttt{C} & \texttt{T} & \texttt{A} & \texttt{C} \\ \hline \end{array} \quad \text{é uma subseqüência de} \quad Y = \begin{array}{|c|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} & \multicolumn{1}{c}{7} \\ \hline \texttt{G} & \underline{\texttt{C}} & \underline{\texttt{T}} & \texttt{G} & \underline{\texttt{A}} & \texttt{G} & \underline{\texttt{C}} \\ \hline \end{array}$$

com índices $2 < 3 < 5 < 7$.

Um subseqüência é **palíndromo** se é a mesma ao ser lida da esquerda para a direita ou da direita para a esquerda. Por exemplo, a seqüência

$$Y = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} & \multicolumn{1}{c}{7} & \multicolumn{1}{c}{8} & \multicolumn{1}{c}{9} & \multicolumn{1}{c}{10} & \multicolumn{1}{c}{11} & \multicolumn{1}{c}{12} & \multicolumn{1}{c}{13} & \multicolumn{1}{c}{14} \\ \hline \texttt{A} & \texttt{C} & \texttt{G} & \texttt{T} & \texttt{G} & \texttt{T} & \texttt{C} & \texttt{A} & \texttt{A} & \texttt{A} & \texttt{A} & \texttt{T} & \texttt{C} & \texttt{G} \\ \hline \end{array}$$

tem várias subseqüências palíndromos, inclusive

$$X = \begin{array}{|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} \\ \hline \texttt{A} & \texttt{C} & \texttt{G} & \texttt{C} & \texttt{A} \\ \hline \end{array} \quad \text{e} \quad X = \begin{array}{|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} \\ \hline \texttt{A} & \texttt{A} & \texttt{A} & \texttt{A} \\ \hline \end{array}\,.$$

Escreva um algoritmo Palindromo-Max(Y, n) que receba uma seqüência $Y[1\,..\,n]$, e devolva o maior comprimento de um subseqüência palíndromo de Y. Seu algoritmo deve consumir tempo $O(n^2)$. Explique sucintamente porque seu algoritmo está correto e tem o consumo de tempo pedido.