ESTRUTURAS DE DADOS - 1o. período de 2009

Gabarito da Primeira Avaliação à Distância

1. (1,5) Sejam dois algoritmos A e B, que resolvem um mesmo problema. O algoritmo A possui complexidade $O(n)$, enquanto que a complexidade de B é $O(n^2)$. Suponha que, neste caso, são válidas as seguintes condições adicionais:

   - As complexidades representam *exatamente* a quantidade de passos efetuada pelos algoritmos para processar n registros.
   - Os algoritmos são executados por um computador que processa cada passo em exatamente um milésimo de segundo.

   Pede-se:
   a. Determinar a quantidade de registros que cada um dos algoritmos pode efetuar em 1 minuto de processamento.

   Resposta: A capacidade de processamento do computador é de $6 \cdot 10^4$ passos por minuto. Logo, o algoritmo A pode efetuar $6 \cdot 10^4$ registros, e o algoritmo B, $\sqrt{6} \cdot 10^2$ registros.

   b. Repetir o cálculo, supondo agora 10 minutos de processamento, ao invés de 1 minuto.

   Resposta: Em 10 minutos, o computador processa $6 \cdot 10^5$ passos. Logo, o algoritmo A pode efetuar $6 \cdot 10^5$ registros, e o algoritmo B, $\sqrt{60} \cdot 10^2$ registros.

2. (1,5) Para cada item abaixo, responda "certo", "errado" ou "nada se pode concluir". Justifique.

   a. Se um algoritmo A possui complexidade polinomial no tamanho da entrada, e um outro algoritmo B possui complexidade exponencial, então o tempo total de processamento para execução do algoritmo A é sempre menor do que o de B, para qualquer instância.

      Resposta: Errado. Consideremos, por exemplo, que o número de passos de A seja $10 \cdot n^{10}$, para qualquer entrada, e que o número de passos de B seja 10^n, para qualquer entrada. Então, o tempo de processamento para execução de A só será menor do que o de B para entradas de tamanho $n \geq 12$.

   b. Se um limite inferior para um problema P é n^2, então qualquer algoritmo ótimo para P tem complexidade de melhor caso $O(n^2)$ ou maior.

      Resposta: Errado. Se um limite inferior para P é n^2, então a complexidade de pior caso de qualquer algoritmo para P (ótimo ou não) é $\Omega(n^2)$. Nada podemos afirmar a respeito da complexidade de melhor caso de qualquer algoritmo para P.

c. Sejam f e g funções tais que $O(f) = O(g)$ e $\Omega(f) = \Omega(g)$. Então podemos concluir que f e g são funções idênticas.

Resposta: Errado. Por exemplo, $f = 3n^2$ e $g = 2n^2$ não são idênticas, e $O(f) = O(g)$ e $\Omega(f) = \Omega(g)$.

3. (1,5) Seja $f_1, f_2, \ldots, f_n$ uma sequência de elementos definida do seguinte modo:

$$f_1 = 0, f_2 = 2, f_3 = 4,$$
$$f_j = f_{j-1} - f_{j-2} + 2f_{j-3} \text{ para } j > 3.$$

Escrever dois algoritmos para determinar o elemento f_n da sequência, o primeiro recursivo e o segundo não recursivo. Calcule a complexidade de cada um, em função de n.

Resposta:

Algoritmo recursivo:

função *seq*(j)
    se $j = 1$ então
        retornar 0
    senão se $j = 2$ então
            retornar 2
        senão se $j = 3$ então
                retornar 4
            senão
                retornar $(seq(j-1) - seq(j-2) + 2 \cdot seq(j-3))$;

Chamada externa: $seq(n)$

Complexidade: É dada pela seguinte equação de recorrência:

$$T(1) = T(2) = T(3) = 1$$
$$T(j) = T(j-1) + T(j-2) + T(j-3)$$

Resolvendo esta recorrência, verificamos que a complexidade deste algoritmo é $O(3^n)$.

Algoritmo iterativo:

$f[1] := 0;$
$f[2] := 2;$
$f[3] := 4;$
para $j = 4 \ldots n$ faça
    $f[j] := f[j-1] - f[j-2] + 2f[j-3];$

A complexidade do algoritmo acima é $O(n)$.

4. (2,0) Determinar a expressão da complexidade média de uma busca linear ORDENADA de 10 chaves, nas seguintes condições:

(i) As probabilidades de busca das chaves de ordem ímpar são iguais entre si.
(ii) As probabilidades de busca das chaves de ordem par são iguais entre si.
(iii) A probabilidade de busca de uma chave de ordem ímpar é a metade da probabilidade de uma chave de ordem par.
(iv) A probabilidade de a chave procurada se encontrar na lista é igual a 50%.

Resposta:

Como a busca se dá em uma lista ordenada, temos 21 entradas distintas (10 entradas em que a chave é encontrada e 11 entradas correspondentes a fracasso). Sejam $E_1, \cdots, E_{10}$ as entradas correspondentes ao sucesso e $E'_0, \cdots, E'_{10}$ as entradas correspondentes ao fracasso (representando os "espaços" entre as chaves da lista).

Considerando a probabilidade de sucesso, temos:

$p(E_1) = p(E_3) = \cdots = p(E_9) = p$

$p(E_2) = p(E_4) = \cdots = p(E_{10}) = 2p$

Como a probabilidade de sucesso é de 50%, temos:

$p(E_1) + p(E_3) + \cdots + p(E_9) + p(E_2) + p(E_4) + \cdots + p(E_{10}) = \frac{1}{2}$

$5 \cdot p + 5 \cdot 2p = \frac{1}{2}$

$p = \frac{1}{30}$

Considerando a probabilidade de fracasso, temos:

$p(E'_0) + p(E'_1) + \cdots + p(E'_{10}) = \frac{1}{2}$

Assumindo que as probabilidades de $E'_0, \cdots, E'_{10}$ são iguais entre si, temos:

$p(E'_i) = \frac{1}{22}, \quad 0 \leq i \leq 10$

O número de passos necessários para cada entrada é:

$t(E_i) = i \ , \qquad 1 \leq i \leq 10$
$t(E'_i) = i + 1 \ , \quad 0 \leq i \leq 10$

Logo, a expressão da complexidade média é dada por:

$$\begin{aligned}
C.M. &= \sum_{i=1}^{10} p(E_i)\, t(E_i) + \sum_{i=0}^{10} p(E'_i)\, t(E'_i) \\
&= \frac{1}{30}(1+3+5+7+9) + \frac{2}{30}(2+4+6+8+10) + \frac{1}{22}\sum_{i=0}^{10}(i+1) \\
&= \frac{25}{30} + 2 + 3 \quad \approx \quad 5,83
\end{aligned}$$

5. (CANCELADA) Escrever um algoritmo de inserção em LISTAS CIRCULARES ENCADEADAS ORDENADAS.

6. (2,0) Seja L uma lista linear que contém os nós $s_1, \ldots, s_n$. Escrever um algoritmo para construir a lista L', cujos nós são $s_1, s_n, s_2, s_{n-1}, \ldots, s_n, s_1$. Isto é, os nós de ordem ímpar de L' correspondem aos nós de L, enquanto que os de ordem par representam a ordem inversa de L.
   O algoritmo descrito deve obedecer à restrição de possuir complexidade igual a $O(n)$.
   Descrever os tipos de estruturas de dados utilizadas.

   Resposta: É utilizada uma pilha P, para armazenar os elementos de L na ordem inversa.

   <u>Algoritmo:</u>

   $topo := 0$
   $pont := ptlista \uparrow .prox$ % ponteiro para a lista L
   enquanto $pont \neq \lambda$ faça
   $\quad topo := topo + 1$
   $\quad P[topo] := pont \uparrow .info$
   $\quad pont := pont \uparrow .prox$

   $ocupar(ptnovo)$ % a lista resultante L' iniciará em $ptnovo$
   $ptaux := ptnovo$
   $pont := ptlista \uparrow .prox$
   enquanto $pont \neq \lambda$ faça
   $\quad ocupar(pt1)$
   $\quad pt1 \uparrow .info := pont \uparrow .info$
   $\quad pont := pont \uparrow .prox$
   $\quad ptaux \uparrow .prox := pt1$
   $\quad ptaux := pt1$
   $\quad ocupar(pt2)$
   $\quad pt2 \uparrow .info := P[topo]$
   $\quad topo := topo - 1$
   $\quad ptaux \uparrow .prox := pt2$
   $\quad ptaux := pt2$
   $\quad ptaux \uparrow .prox := \lambda$

7. (1,5) Seja $1, 2, \ldots, n$ uma sequência de elementos que serão inseridos e retirados de uma pilha P uma vez cada. A ordem de inclusão dos elementos na pilha é fixa, sendo igual a $1, 2, \ldots, n$. Por outro lado, a ordem de remoção não o é. A sequência de remoção define uma permutação que representa o movimento dos nós na pilha. Por exemplo, com $n = 3$, a sequência de operações "incluir em P, incluir em P, retirar de P, incluir em P, retirar de P, retirar de P" produzirá a permutação $2, 3, 1$ a partir da entrada $1, 2, 3$. Representando por I, R, respectivamente, as operações de inserção e remoção da pilha, a permutação $2, 3, 1$ pode ser denotada por $IIRIRR$. De um modo geral, uma permutação é chamada *admissível* quando ela puder ser obtida mediante uma sucessão de inclusões e remoções

em uma pilha a partir da permutação $1, 2, \ldots, n$. Assim, por exemplo, a permutação $2, 3, 1$ é admissível. Pede-se:

(i) Determinar a permutação correspondente a $IIIRIRRIRR$, $n = 5$.
Resposta: $3, 4, 2, 5, 1$.

(ii) A permutação $n, n-1, \ldots, 1$ é sempre admissível ? Justificar.
Resposta: Sim. Esta permutação corresponde a n inclusões, seguidas de n remoções. Para $n = 4$, por exemplo, temos a sequência $IIIIRRRR$.

(iii) A permutação $1, n, 2, n-1, \ldots, (n+1)/2$, com n ímpar, é admissível ? Justificar.
Resposta: Não. Esta permutação só é admissível para $n \leq 3$. Para qualquer $n \geq 4$ (ímpar ou não), a sequência $p_i, p_j, p_k = n, 2, n-1$ não é admissível, pois $p_j < p_k < p_i$.