Estrutura de Dados - 2o. período de 2005

Gabarito da Primeira Avaliação à Distância

1. Escrever as seguintes funções em notação O:
   $n^3 - 1$; $n^2 + 2\log n$; $3n^n + 5.2^n$; $(n-1)^n + n^{n-1}$; 302.

   Resposta: $n^3 - 1 = O(n^3)$; $n^2 + 2\log n = O(n^2)$; $3n^n + 5.2^n = O(n^n)$; $(n-1)^n + n^{n-1} = O(n^n)$; $302 = O(1)$.

2. Para cada item abaixo, responda "certo" ou "errado", justificando:

   a. Se a complexidade de melhor caso de um algoritmo for f, então o número de passos que o algoritmo efetua, qualquer que seja a entrada, é $\Omega(f)$.

   Resposta: certo, pois o melhor caso corresponde ao mínimo de passos que o algoritmo realiza.

   b. Se a complexidade de pior caso de um algoritmo for f, então o número de passos que o algoritmo efetua, qualquer que seja a entrada, é $\Theta(f)$.

   Resposta: errado. O certo seria dizer que o algoritmo efetua $O(f)$ passos qualquer que seja a entrada, uma vez que o pior caso corresponde ao número máximo de passos.

   c. A complexidade de melhor caso de um algoritmo para um certo problema é necessariamente maior do que qualquer limite inferior para o problema.

   Resposta: errado, pois o limite inferior refere-se apenas à complexidade de pior caso.

3. A *seqüência de Fibonacci* é uma sequência de elementos $f_1, f_2, \ldots, f_n$, definida do seguinte modo: $f_1 = 0, f_2 = 1, f_j = f_{j-1} + f_{j-2}$. Elaborar um algoritmo, não recursivo, para determinar o elemento f_n da sequência, cuja complexidade seja linear em n.

   Resposta:

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

4. Determinar a expressão da complexidade média de uma busca não ordenada de n chaves, n par, em que as probabilidades de busca das chaves de ordem ímpar são iguais entre si, sendo esse valor igual ao dobro da probabilidade de qualquer chave par. Supor, ainda, que a probabilidade de a chave se encontrar na lista é igual a q.

   Resposta: Seja p a probabilidade de uma chave ímpar. Temos então que a probabilidade de uma chave par é $p/2$. Distribuindo a probabilidade q pelas n chaves, temos que

   $$(n/2)p + (n/2)(p/2) = q.$$

   Concluímos que $p = 4q/3n$. Logo, a expressão da complexidade média neste caso é dada pela expressão:

$$C.M. = (1 + 3 + \ldots + n - 1)\frac{4q}{3n} + (2 + 4 + \ldots + n)\frac{2q}{3n} + n(1 - q)$$

onde a parcela final $n(1-q)$ refere-se ao caso em que a informação procurada não se encontra na lista.

5. Comparar algoritmos de busca, inserção e remoção em uma lista ordenada nas alocações sequencial e encadeada.

   Resposta:

   Busca - $O(n)$ tanto para alocação sequencial como para encadeada.
   Inserção - $O(n)$ para alocação sequencial, $O(1)$ para alocação encadeada.
   Remoção - $O(n)$ para alocação sequencial, $O(1)$ para alocação encadeada.

   Obs: na prática, a inserção e a remoção exigem uma busca prévia. Portanto, na prática, todos os algoritmos acima se tornam $O(n)$.

6. Sejam duas listas, ordenadas, simplesmente encadeadas com nó-cabeça. Apresentar um algoritmo que intercale as duas listas de forma que a lista resultante esteja também ordenada.

   Resposta:

   $pont1 := ptlista1 \uparrow .prox$ % ponteiro para a lista 1
   $pont2 := ptlista2 \uparrow .prox$ % ponteiro para a lista 2
   $ptaux := ptlista1$ % a lista resultante iniciará em $ptlista1$

   enquanto $pont1 \neq \lambda$ e $pont2 \neq \lambda$ faça
       se $pont1 \uparrow .info < pont2 \uparrow .info$
           então
               $ptaux \uparrow .prox := pont1$
               $ptaux := pont1$
               $pont1 := pont1 \uparrow .prox$
           senão
               $ptaux \uparrow .prox := pont2$
               $ptaux := pont2$
               $pont2 := pont2 \uparrow .prox$
   fim-enquanto

   se $pont1 = \lambda$
       então $ptaux \uparrow .prox := pont2$
       senão $ptaux \uparrow .prox := pont1$

7. Descreva algoritmos de inserção e remoção em pilhas e filas, implementadas utilizando alocação encadeada.

   Resposta:

Inserção na pilha:
ocupar(pt)
$pt \uparrow .info := novo - valor$
$pt \uparrow .prox := topo$
$topo := pt$

Remoção da pilha:
se $topo \neq \lambda$ então
    $pt := topo$
    $topo := topo \uparrow .prox$
    $valor - recuperado := pt \uparrow .info$
    $desocupar(pt)$
senão $underflow$

Inserção na fila:
$ocupar(pt)$ $pt \uparrow .info := novo - valor$
$pt \uparrow .prox := \lambda$
se $fim \neq \lambda$ então
    $fim \uparrow .prox := pt$
senão $inicio := pt$
$fim := pt$

Remoção da fila:
se $inicio \neq \lambda$ então
    $pt := inicio$
    $inicio := inicio \uparrow .prox$
    se $inicio = \lambda$ então $fim := \lambda$
    $valor - recuperado := pt \uparrow .info$
    $desocupar(pt)$
senão $underflow$

8. Seja $1, 2, \ldots, n$ uma seqüência de elementos que serão inseridos e posteriormente retirados de uma pilha P uma vez cada. A ordem de inclusão dos elementos na pilha é $1, 2, \ldots, n$, enquanto a de remoção depende das operações realizadas. Por exemplo, com $n = 3$, a seqüência 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 $IIIRRIRR$, $n = 4$.
   Resposta: $3, 2, 4, 1$

   (ii) Dê um exemplo de permutação não admissível.
   Resposta: $4, 1, 2, 3$ (para $n = 4$). Motivo: após remover o 4, o 1 se encontra no fundo da pilha e não pode ser o próximo a ser removido.