Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
Gabarito da AP1 - Primeiro Semestre de 2009

Nome -
Assinatura -

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

---

1. (Valor 1,5) Explicar, com precisão os seguintes conceitos:

   - Complexidade de caso médio.

     Resposta: Sejam A um algoritmo, $E = \{E_1, \cdots, E_m\}$ o conjunto de todas as entradas possíveis de A e t_i o número de passos efetuados por A, quando a entrada for E_i. A complexidade de caso médio é definida por $\sum_{1 \leq i \leq m} p_i t_i$, onde p_i é a probabilidade de ocorrência da entrada E_i.

   - Notação O.

     Resposta: Sejam f, h funções reais de variável inteira n. Dizemos que f é $O(h)$ quando existirem constante $c > 0$ e um valor inteiro n_0 tais que: $n > n_0 \Rightarrow f(n) \leq c \cdot h(n)$. A função h atua como limite superior para valores assintóticos da função f.

   - Lista encadeada.

     Resposta: É uma lista cujos nós encontram-se aleatoriamente dispostos na memória e são interligados por ponteiros, que indicam a posição do próximo elemento da tabela.

2. (Valor 2,5) Dadas duas listas L_1 e L_2, simplesmente encadeadas, com n nós cada, descrever um algoritmo para verificar se L_1 representa a inversão de L_2. Isto é, o algoritmo deve verificar se os nós de L_1 são exatamente os nós de L_2, porém em ordem inversa. Pede-se:

   - Descrever a estratégia geral do algoritmo, em palavras.

     Resposta: O algoritmo percorre a lista L_1, armazenando cada elemento em uma pilha P. Dessa forma, os elementos de L_1 são acessados em P na ordem inversa. Em seguida, os elementos de P são comparados um a um com a lista L_2. Caso a comparação seja positiva para todos os n elementos, então L_1 é a inversão de L_2.

   - Descrever o algoritmo, supondo que as listas L_1 e L_2 estão armazenadas com a utilização de ponteiros.

     Resposta:

Algoritmo:

$topo := 0$
$pont1 := ptlista1 \uparrow .prox$
enquanto $pont1 \neq \lambda$ faça
    $topo := topo + 1$
    $P[topo] := pont1 \uparrow .info$
    $pont1 := pont1 \uparrow .prox$

$igual := verdadeiro$
$pont2 := ptlista2 \uparrow .prox$
enquanto ($igual$ e $pont2 \neq \lambda$) faça
    se $P[topo] = pont2 \uparrow .info$ então
        $topo := topo - 1$
        $pont2 := pont2 \uparrow .prox$
    senão
        $igual := falso$

se $igual$ então
    imprimir ("L_1 representa a inversão de L_2")
senão
    imprimir ("L_1 não representa a inversão de L_2")

- Determinar e justificar a complexidade do algoritmo.
  Resposta: A complexidade do algoritmo é $\theta(n)$, uma vez que percorre as listas L_1 e L_2 exatamente uma vez, e executa um número constante de passos para cada elemento das listas.

3. (Valor 2,5) Seja uma estrutura de dados E composta por duas filas F_1 e F_2, que compartilham a mesma área de tamanho correspondente a n nós. No caso, F_1 e F_2 compartilham o mesmo vetor de n elementos, com F_1 se desenvolvendo sequencialmente da extremidade esquerda do vetor para a direita, enquanto que F_2 ocupa as posições a partir da extremidade direita e se desenvolve, em sequência, para a esquerda. Pede-se:

   - Formular um algoritmo para inserir dados nesta estrutura. A entrada deste algoritmo consiste da estrutura E, do dado a ser in-

serido e da informação em qual fila F_1 ou F_2 deve ser realizada a inserção.

Resposta: Sejam $ini1$ e $ini2$ ($fim1$ e $fim2$) as variáveis que apontam para o início (fim) das filas F_1 e F_2, respectivamente. Inicialmente, temos $ini1 = ini2 = fim1 = fim2 = 0$, indicando que F_1 e F_2 estão vazias. Seja b uma variável booleana que indica em qual fila o dado será inserido. $b = verdadeiro$ indica inserção na fila F_1.

Algoritmo:

```
se fim2 = (fim1 + 1) então
      overflow
senão
      se b então
            se ini1 = 0 então        % F1 está vazia
                  ini1 := 1, fim1 := 1
            senão
                  fim1 := fim1 + 1
            E[fim1] := novo-valor
      senão
            se ini2 = 0 então        % F2 está vazia
                  ini2 := n, fim2 := n
            senão
                  fim2 := fim2 − 1
            E[fim2] := novo-valor
```

- Descrever as condições de overflow e underflow na estrutura.

  Resposta: Ocorre overflow quando $fim2 = (fim1 + 1)$ e tentamos inserir um dado em F_1 ou em F_2. Ocorre underflow quando $ini1 = fim1 = 0$ e tentamos remover um dado de F_1, ou quando $ini2 = fim2 = 0$ e tentamos remover um dado de F_2.

- Determinar e justificar a complexidade da inserção.

  Resposta: A complexidade da inserção é $O(1)$, já que é executado um número constante de passos.

4. (Valor 1,5) Dê as definições de:

   - árvore binária completa

     Resposta: Uma árvore binária completa é aquela em que, se v é um nó tal que alguma subárvore de v é vazia, então v se localiza ou no último ou no penúltimo nível da árvore.

   - árvore binária cheia

     Resposta: Uma árvore binária cheia é aquela em que, se v é um nó com alguma de suas subárvores vazias, então v se localiza no último nível.

   - árvore estritamente binária

     Resposta: Uma árvore estritamente binária é uma árvore binária em que cada nó possui 0 ou 2 filhos.

5. (Valor 2,0) Escrever o algoritmo de busca binária em uma lista ordenada com chaves $s_1, \ldots, s_n$, sendo $s_1 < ... < s_n$. Em seguida, calcular exatamente o número de comparações entre chaves que o algoritmo realiza, supondo $n = 33$ e que a chave procurada esteja na décima posição, da esquerda para a direita.

   Resposta:

   função *busca-bin*(L, i, f, x)
   se $i > f$ então *retornar* -1
   senão
   $meio := \lfloor (i + f)/2 \rfloor$
   se $L[meio] = x$ então *retornar* $meio$
   senão
   se $L[meio] < x$ então *retornar* *busca-bin*$(L, meio + 1, f, x)$
   senão *retornar* *busca-bin*$(L, i, meio - 1, x)$

   São realizadas 4 comparações.