



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Estrutura de Dados**

**AP1 - 2019/1**

**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) Leia atentamente o texto a seguir, e depois resolva as questões abaixo.

Sabemos que um problema computacional pode admitir vários algoritmos para resolvê-lo, que podem ter complexidades de pior caso diferentes. Dentre esses algoritmos, aqueles que possuem as menores complexidades de pior caso podem ou não ser ótimos, pois o que caracteriza um algoritmo ótimo não é o fato de ele ser "melhor" do que os outros, mas é algo relacionado à complexidade intrínseca do problema que ele resolve, isto é, ao limite inferior do problema.

(a) (1,0) Dê a definição formal de algoritmo ótimo.

(b) (1,0) Responda: Um problema computacional pode admitir mais de um algoritmo ótimo para resolvê-lo? Justifique.

2) Considere um volumoso cadastro de CPF's, no qual as seguintes operações são executadas:

1. Busca de um CPF no cadastro.
2. Inserção de um novo CPF no cadastro.
3. Remoção de um CPF do cadastro.

Suponha ainda que, para cada CPF, existe um contador que é incrementado de uma unidade a cada vez que este é buscado no cadastro. Isto forma uma estatística dos acessos.

Temos a seguir várias estruturas de dados que poderiam ser utilizadas para implementar o cadastro de CPF's:

(a) Lista linear não ordenada;

- (b) Lista linear ordenada por CPF;
- (c) Lista linear ordenada (decreasingmente) por contador;
- (d) Lista simplesmente encadeada não ordenada;
- (e) Lista simplesmente encadeada ordenada pelo CPF;
- (f) Lista simplesmente encadeada ordenada (decreasingmente) pelo contador.

Resolva as questões a seguir, justificando:

- (a) (1,5) Quais as melhores estruturas em relação à operação 1?
- (b) (1,5) Quais as melhores estruturas em relação à operação 2?
- (c) (1,5) Quais as melhores estruturas em relação à operação 3?

3) Dado um vetor  $V$  com  $n > 1$  elementos distintos, a mediana de  $V$  é um elemento  $V[i]$  (para algum índice  $i$  entre 1 e  $n$ ) com a seguinte propriedade: existem  $\text{int}(n/2)$  elementos em  $V$  que são menores do que  $V[i]$ , onde  $\text{int}(x)$  é a parte inteira do número  $x$ . Como exemplo, considere o vetor  $V = (2\ 19\ 26\ 3\ 4\ 1\ 5)$ . Temos  $n=7$  e  $\text{int}(n/2)=3$ . Portanto, a mediana de  $V$  é o elemento  $V[5] = 4$ .

(a) (1,5) Escreva um algoritmo que monta um outro vetor  $M$ , contendo  $n$  elementos, com a seguinte propriedade: para cada índice  $i$  entre 1 e  $n$ , o elemento  $M[i]$  armazena quantos elementos de  $V$  são menores do que  $V[i]$ . Como exemplo, para o vetor  $V$  acima, temos que  $M = (1\ 5\ 6\ 2\ 3\ 0\ 4)$ .

(b) (1,0) Suponha que o vetor  $M$  já esteja calculado. Escreva um comando do tipo “enquanto” que obtém a mediana de  $V$  a partir de  $M$ . Se possível, faça o comando se encerrar tão logo a mediana seja encontrada.

4) (1,0) Considere uma fila  $F$  contendo as posições de 1 a 5. A variável  $f$  marca a posição de início da fila (“frente”), e a variável  $r$  marca a posição de fim da fila (“retaguarda”). No início, a fila  $F$  encontra-se vazia, e as variáveis  $f$  e  $r$  valem zero.

Usamos a notação  $R$  para denotar a operação de remoção de um elemento da fila  $F$ , e a notação  $I(X)$  para denotar a operação de inserção de um elemento  $X$  na fila  $F$ .

Considere a seguinte sequência de operações em  $F$ :

$I(A), I(B), I(C), R, I(D), R, I(E), I(G), I(H), R, R, R, R, I(J)$

Desenhe como fica a fila  $F$  após a sequência de operações acima, e forneça os valores finais das variáveis  $f$  e  $r$ . Use um traço (-) para denotar as posições vazias. Como um exemplo de configuração, poderíamos ter:  $F = (-\ -\ C\ D\ -)$ , com  $f=3$  e  $r=4$ .