Algoritmos e Estruturas de Dados II

Exercícios dos Quizzes até a 2ª Prova

Questões extraídas dos *quizzes* da disciplina elaborados pela Prof.ª Eveline Alonso
Resoluções elaboradas pelo aluno Luca Ferrari Azalim

---

1. Analise as seguintes afirmativas sobre métodos de ordenação:

I. O Quicksort divide um conjunto de itens em conjuntos menores, que são ordenados de forma independente, e depois os resultados são combinados para produzir a solução de ordenação do conjunto maior.

II. Seleção é um método que consiste em selecionar o menor item de um vetor e trocá-lo com o item que estiver na primeira posição. Essas duas operações são repetidas com os itens restantes até o último elemento.

III. O Heapsort é uma extensão do método de ordenação por inserção, contornando o problema que ocorre quando o menor item de um vetor está na posição mais à direita.

Assinale a alternativa CORRETA:

a) As afirmativas I, II e III estão corretas.
b) A afirmativa I está errada e as afirmativas II, III estão corretas.
c) A afirmativa III está errada e as afirmativas I, II estão corretas.
d) A afirmativa II está errada e as afirmativas I, III estão corretas.

---

2. Sobre o algoritmo de ordenação Heapsort, assinale a afirmação correta:

a) Seu desempenho de pior caso é menor do que o do algoritmo Mergesort.
b) A estrutura de dados que utiliza, chamada heap, pode ser implementada por meio de um vetor.
c) Seu desempenho de pior caso é pior do que o do algoritmo Quicksort.
d) Seu desempenho de pior caso é o mesmo da ordenação por inserção.
e) Utiliza ordenação por árvore de decisão, ao invés de ordenação por comparação.

---

3. Um engenheiro de software construiu uma função para ordenar vetores de inteiros por meio do algoritmo de ordenação por inserção. A versão iterativa deste algoritmo possui dois loops aninhados. Suponha que esse engenheiro de software tenha inserido, imediatamente antes do incremento da variável de controle do loop mais externo, uma chamada a uma função que

percorre e exibe o conteúdo do vetor que está sendo ordenado. O trecho de código a seguir ilustra como essa chamada é feita.

```
for(int i = 1; i < vetor.length; i++) {

    /*
     * Os demais comandos que implementam o algoritmo de ordenação
     * por inserção devem substituir estas linhas de comentário.
     */

    exibeVetor(vetor);

}
```

O vetor abaixo representa o vetor que foi passado como parâmetro em uma chamada da função de ordenação acima.

78	**12**	**35**	**1**	**17**	**4**	**43**	**11**	**17**	**1**

O que será exibido quando o valor da variável “i” for igual a 3?

a) 1 1 4 78 17 35 43 11 17 12
b) 1 12 4 17 11 17 1 35 43 78
c) 1 12 35 78 17 4 43 11 17 1
d) 1 1 11 12 17 4 17 35 43 78
e) 1 1 4 11 17 35 43 78 17 12

---

4. Seja a seguinte sequência numérica [15, 11, 16, 18, 23, 5, 10, 22, 21, 12], que deve ser ordenada por meio da aplicação de uma variação do algoritmo de ordenação por seleção clássico. Nessa variação, a ordenação ocorre a partir da última posição do vetor em direção a seu início. Qual alternativa abaixo corresponde à sequência, parcialmente ordenada, encontrada depois de completada a quinta passagem do algoritmo?

a) [12, 11, 5, 10, 15, 16, 18, 21, 22, 23]
b) [15, 11, 16, 18, 12, 5, 10, 21, 22, 23]
c) [15, 11, 16, 10, 12, 5, 18, 21, 22, 23]
d) [10, 11, 5, 12, 15, 16, 18, 21, 22, 23]
e) [15, 11, 5, 10, 12, 16, 18, 21, 22, 23]

---

5. A ordenação de elementos em um vetor pode ser executada a partir de diversos algoritmos conhecidos e que são adequados para situações específicas. Sobre algoritmos de ordenação, dadas as seguintes afirmativas,

1. O algoritmo Bubblesort é ineficiente para ordenar muitos itens.
2. O algoritmo de ordenação por seleção, para ordenação crescente, consiste em mover o menor valor do vetor para a primeira posição, depois o segundo menor para a segunda posição e assim sucessivamente até os dois últimos valores.
3. O algoritmo Quicksort ordena os valores de um vetor por meio de sucessivas seleções do elemento correto a ser posicionado em um segmento já ordenado.

Verifica-se que está(ão) correta(s):

a) 2, apenas.
b) 1, apenas.
c) 1 e 2, apenas.
d) 1, 2 e 3.
e) 1 e 3, apenas

---

6. Analise o algoritmo de ordenação que se segue:

```
procedimento ordena(A[]: inteiro; n: inteiro)

var i: inteiro;
var j: inteiro;
var aux: inteiro;

início

    para i <- 0 até (n - 1), com passo 1, faça

        para j <- 0 até (n - 2), com passo 1, faça

            se (A[j] > A[j + 1]) então

                aux <- A[j];

                A[j] <- A[j + 1];

                A[j + 1] <- aux;

            fim_se

        fim_para;
```

```
    fim_para
fim
```

Considere que o vetor “A” foi inicialmente preenchido com os seguintes elementos, nessa ordem: 16, 18, 15, 13, 36.

Assinale a alternativa que indica o nome do método de ordenação descrito pelo algoritmo acima e a quantidade de trocas entre elementos realizadas ao se ordenar o vetor “A” por meio do emprego desse método de ordenação.

a) Ordenação por inserção, com 4 trocas.
b) Bubblesort, com 5 trocas.
c) Bubblesort, com 4 trocas.
d) Ordenação por seleção, com 5 trocas.
e) Ordenação por seleção, com 4 trocas.

---

7. Um heap (fila de prioridades) é uma estrutura de dados muito importante, que tem duas utilidades principais: organizar o acesso a um recurso com base na prioridade dos requerentes (processos, impressões, etc); ou servir como base de um algoritmo de ordenação muito eficiente denominado heapsort. Para poder servir a esses propósitos, um heap possui uma série de propriedades especiais que têm que ser mantidas por todas as operações nele realizadas. Levando em consideração estas propriedades, analise as afirmativas abaixo:

I. 50 40 49 39 45 46 representa um heap sintaticamente correto.

II. Dado o heap 21 14 10 9 5, após a inserção do elemento 12 a configuração desse heap será: 21 14 12 9 5 10

III. Dado o heap 21 14 10 9 5, a retirada do elemento do topo se dará por meio dos seguintes passos: 5 14 10 9 -> 14 5 10 9 -> 14 9 10 5

É correto APENAS o que se afirma em:

a) I e II
b) II
c) III
d) II e III

e) I

---

8. O Quicksort é considerado o algoritmo de ordenação baseado em comparação mais eficiente, mas em alguns casos sua complexidade é igual ao do Bubblesort. Assinale a alternativa que indica a complexidade que o Quicksort pode atingir quando o vetor está ordenado em ordem decrescente:

a) O(nˆ2)
b) O(n)
c) O(n log n)
d) O(nˆ2 log n)
e) O(log n)

---

9. Analise o algoritmo de ordenação que se segue:

```
procedimento ordena(A[]: inteiro; n: inteiro)

var i: inteiro;
var j: inteiro;
var temp: inteiro;
var pos: inteiro;

início

    para i <- 0 até (n - 2), com passo 1, faça

        pos <- i;

        para j <- (i + 1) até (n - 1), com passo 1, faça

            se (A[pos] < A[j]) então

                pos <- j;

            fim_se

        fim_para;

        temp <- A[i];

        A[i] <- A[pos];

        A[pos] <- temp;
```

```
    fim_para

fim
```

Assinale a alternativa que indica o nome do método de ordenação descrito pelo algoritmo acima e se a ordenação do vetor “A” ocorre de forma crescente ou decrescente.

a) Ordenação por inserção, ordenação decrescente.
b) Bubblesort, ordenação crescente.
c) Ordenação por seleção, ordenação crescente.
d) Ordenação por seleção, ordenação decrescente.
e) Ordenação por inserção, ordenação crescente.

---

10. A maioria dos softwares de aplicação possui comandos de "Desfazer" e "Refazer". O primeiro desfaz a última operação ou texto digitado, enquanto que, o segundo refaz uma operação ou texto desfeito, conforme sugerem os nomes dos comandos.

Internamente, nos softwares, podem ser usadas duas estruturas de dados que armazenam as sucessivas operações de "Desfazer" e "Refazer", de modo que o próximo "Refazer" sempre recupera o último "Desfazer". Os tipos de estrutura de dados que podem ser usados para "Desfazer" e "Refazer" são, respectivamente:

a) Pilha e fila
b) Pilha e pilha
c) Fila e pilha
d) Fila e fila

---

11. Considere o tipo abstrato de dados Pilha e suas típicas operações: criarPilha(), vazia(), empilhar(item), desempilhar() e copiar().

Analise o pseudocódigo abaixo e assinale a alternativa correta:

```
função func(A[]: inteiro; n: inteiro): booleano

var i: inteiro
var pilhaEntrada: Pilha
var pilhaCopia: Pilha;
var pilhaInverso: Pilha;
```

```
início

   pilhaEntrada = criarPilha();
   pilhaInverso = criarPilha();

   para i <- 0 até (n - 1), com passo 1, faça
      pilhaEntrada.empilhar(A[i]);
   fim_para

   pilhaCopia = pilhaEntrada.copiar();

   enquanto (!pilhaCopia.vazia()) faça
      pilhaInverso.empilhar(pilhaCopia.desempilhar());
   fim_enquanto

   enquanto (!pilhaEntrada.vazia()) faça
      se (pilhaInverso.desempilhar() != pilhaEntrada.desempilhar()) então

         return falso;

      fim_se
   fim_enquanto

   return verdadeiro;

fim
```

a) A execução de “func” pode acessar posições de memória não alocadas anteriormente.

b) Se a variável “pilhaCopia”, do algoritmo descrito acima, fosse retirada, juntamente com todo o pseudocódigo que a referencia; nenhuma alteração seria observada na execução da função, pois pilhaCopia é preenchida e, em seguida, esvaziada.

c) A função “func” verifica se o vetor de entrada “A” é simétrico em relação à posição do meio, ou seja, se ele apresenta a mesma sequência de inteiros quer seja percorrido de frente para trás ou de trás para frente.

d) A execução da função “func” sempre retornará verdadeiro, quaisquer que sejam os parâmetros de entrada.

e) O TAD Pilha não é o mais adequado para a resolução desse problema. Deveria ter sido utilizado o TAD Fila para a resolução mais eficaz do problema proposto.

---

12. A pilha é uma estrutura de dados que permite a inserção e a remoção de dados sempre por meio de regras pré-definidas. Para que essas operações sejam realizadas, são utilizados dois métodos: push() e pop().

Com base nessas informações, considere que um programa possua uma pilha p, inicialmente vazia, e que as seguintes operações foram realizadas, nesta ordem: p.push(10); p.push(5); p.push(3); p.push(50); p.pop(); p.push(11); p.push(9); p.push(20); p.pop(); p.pop().

Ao fim da execução desses comandos, quais serão o topo da pilha e o somatório dos elementos ainda dentro da pilha, respectivamente?

a) 3 e 29
b) 50 e 68
c) 20 e 58
d) 9 e 38
e) 11 e 29

---

13. O método abaixo deve ser implementado na classe Pilha, que implementa o tipo abstrato de dados Pilha e suas típicas operações básicas: pilhaVazia(), empilhar(item) e desempilhar().

Esse método deve ser capaz de fazer e retornar uma cópia exata da pilha.

```
// Método capaz de fazer e retornar uma cópia exata da pilha.

public Pilha copiar() {

    Pilha pilhaCopia = new Pilha(), pilhaAux = new Pilha();
    Celula aux;

    if (!pilhaVazia()) {

        I

        while (II) {

            pilhaAux.empilhar(new Item(aux.item.getId()));

            III

        }
```

```
        while (!pilhaAux.pilhaVazia()){

            IV

        }

    }

    V

}
```

Analise as alternativas abaixo e indique a única que identifica corretamente os comandos que devem preencher as lacunas, numeradas de I a V.

a) I - aux = topo.proximo; II - aux != fundo; III – aux++; IV - pilhaCopia.empilhar(pilhaAux.desempilhar()); V - return (pilhaCopia);

b) I - aux = topo; II - aux != fundo; III - aux++; IV - pilhaAux.desempilhar(); V - return (aux);

c) I - aux = topo; II - aux != fundo; III - aux = aux.proximo; IV - pilhaCopia.empilhar(pilhaAux.desempilhar()); V - return (pilhaCopia);

d) I - aux = topo.proximo; II - aux != null; III - aux = aux.proximo; IV - pilhaCopia.empilhar(pilhaAux.desempilhar()); V - return (pilhaCopia);

e) I - aux = topo; II - aux != fundo; III - aux = aux.proximo; IV - pilhaAux.empilhar(pilhaCopia.desempilhar()); V - return (pilhaAux);

---

14. Suponha as seguintes operações de empilhar e desempilhar realizadas em uma pilha vazia: empilhar(10), empilhar(5), empilhar(7), desempilhar(), empilhar(13), empilhar(4), desempilhar(), desempilhar(), desempilhar(), empilhar(1), desempilhar(), desempilhar(). Assinale a alternativa que contenha a sequência de números que foram removidos da pilha pela operação de desempilhar na sequência que foi executada.

a) 7 - 4 - 13 - 5 - 1 - 10
b) 10 - 5 - 13 - 7 - 4 - 1
c) 10 - 5 - 7 - 13 - 4 - 1
d) 7 - 4 - 10 - 5 - 13 – 1
e) 5 - 13 - 4 - 10 - 5 - 1

---

15. Qual é a estrutura de dados que permite apenas que novos elementos sejam adicionados na última posição e que elementos sejam retirados na primeira posição?

a) Lista encadeada
b) Lista duplamente encadeada
c) Pilha
d) Fila
e) Lista circular

---

16. O trecho de código abaixo deve ser implementado na classe Fila<E>, que implementa o tipo abstrato de dados Fila e suas típicas operações básicas: vazia(), enfileirar() e desenfileirar().

Ele deve ser executado sobre uma fila não-vazia (“this”).

Considere que o primeiro item da fila (item localizado após a célula sentinela) ocupa a posição 0 da fila. A célula seguinte ocupa a posição 1 e assim sucessivamente.

```
Fila<E> filaAux = new Fila<E>();
Celula<E> celY = this.frente;
Celula<E> celX = celY.getProximo();

while (celX != null) {

    filaAux.enfileirar(celX.getItem());
    celY.setProximo(celX.getProximo());
    celX.setProximo(null);

    if (celY.getProximo() != null) {

        celY = celY.getProximo();
        celX = celY.getProximo();

    } else {

        celX = null;

    }

}
```

```
this.tras = celY;
return (filaAux);
```

O que esse trecho de código faz?

a) Divide a fila original (“this”) em duas, da seguinte maneira: devem permanecer na fila original (“this”) os elementos que ocupam, atualmente, as primeiras posições nessa fila, até sua metade. Devem ser enfileirados, na fila que será retornada, os itens que ocupam, atualmente, as últimas posições da fila original (“this”), a partir do meio dessa.
b) Divide a fila original (“this”) em duas, da seguinte maneira: devem permanecer na fila original (“this”) os elementos que ocupam, atualmente, posição ímpar nesta fila. Devem ser enfileirados, na fila que será retornada, os itens que ocupam, atualmente, posição par na fila original (“this”).

c) Retira da fila original (“this”) os itens que ocupam posição par, retornando essa fila modificada.

d) Retira da fila original (“this”) os itens que ocupam posição ímpar, retornando essa fila modificada.

---

17. José construiu uma estrutura de dados do tipo fila e executou uma sequência de comandos sobre essa fila: enfileirar(1); enfileirar(4); desenfileirar(); enfileirar(2); enfileirar(3); enfileirar(5); enfileirar(6); desenfileirar(). Considere que a fila estava originalmente vazia.

Após a execução dessa sequência de comandos, escolha, entre as alternativas abaixo, a única que contém o conteúdo da fila:

a) 1-2-5-6
b) 3-4-5-6
c) 2-3-5-6
d) 1-2-3-5
e) 1-2-4-5-6

---

18. Considere a estrutura de dados do tipo Fila e suas típicas operações básicas: enfileirar(item) e desenfileirar().

Suponha que cinco documentos estejam em uma fila para impressão “f”, na seguinte ordem: doc1, doc2, doc3, doc4, doc5. No entanto, deseja-se imprimir imediatamente os documentos doc1, doc2 e doc5, nessa sequência, deixando-se doc3 e doc4 na fila “f”, na mesma ordem em que se encontravam originalmente.

Em qual pseudocódigo abaixo a sequência de operações produz o resultado pretendido, com o

mínimo de movimentações de itens na fila “f”?

a) f.desenfileirar(); f.desenfileirar(); f.enfileirar(f.desenfileirar()); f.desenfileirar(); f.enfileirar(f.desenfileirar());

b) f.enfileirar(f.desenfileirar()); f.enfileirar(f.desenfileirar()); f.desenfileirar(); f.desenfileirar(); f.enfileirar(f.desenfileirar());

c) f.desenfileirar(); f.enfileirar(f.desenfileirar()); f.desenfileirar(); f.enfileirar(f.desenfileirar()); f.desenfileirar();

d) f.enfileirar(f.desenfileirar()); f.enfileirar(f.desenfileirar()); f.enfileirar(f.desenfileirar()); f.enfileirar(f.desenfileirar()); f.enfileirar(f.desenfileirar());

e) f.desenfileirar(); f.desenfileirar(); f.enfileirar(f.desenfileirar()); f.enfileirar(f.desenfileirar()); f.desenfileirar();

---

19. Considere que os números 10, 11, 12, 13 e 14 foram inseridos, nessa ordem, em uma fila. Esses mesmos números foram inseridos na mesma ordem em uma pilha. Nesse caso,

a) o topo da pilha é o número 10.
b) o primeiro elemento a ser removido da pilha é o número 10.
c) o número 14 é o primeiro elemento a ser removido da fila.
d) o último elemento a ser removido da fila é o número 14.