

Matéria: Tecnologia da Informação

Assunto: Estruturas de Dados e Algoritmos

Resumo Teórico do Assunto

Para resolver as questões apresentadas sobre Estruturas de Dados e Algoritmos, é fundamental compreender os conceitos de algoritmos de ordenação, especificamente o **Bubble Sort**, e a estrutura de dados **Fila (Queue)**.

Estruturas de Dados e Algoritmos: Fundamentos Essenciais

Estruturas de Dados são formas organizadas de armazenar e gerenciar dados, permitindo acesso e modificação eficientes. **Algoritmos** são sequências finitas de instruções bem definidas e não ambíguas, que resolvem um problema específico ou realizam uma tarefa.

I. Algoritmos de Ordenação

Algoritmos de ordenação são métodos utilizados para organizar elementos de uma lista ou vetor em uma sequência específica (crescente ou decrescente).

1. Bubble Sort (Ordenação por Bolha)

- **Conceito:** O Bubble Sort é um algoritmo de ordenação simples que funciona repetidamente percorrendo a lista, comparando pares de elementos adjacentes e trocando-os de lugar se estiverem na ordem errada. O processo se repete até que nenhum par precise ser trocado, indicando que a lista está ordenada.

- **Funcionamento:**

1. Ele compara o primeiro elemento com o segundo. Se o primeiro for maior que o segundo (para ordenação crescente), eles são **trocados** de posição.
2. Em seguida, compara o segundo com o terceiro, e assim por diante, "borbulhando" o maior elemento para o final da lista a cada passagem completa.
3. Este processo é repetido para cada elemento, diminuindo o número de comparações a cada passagem, pois os maiores elementos já estão no final.

- **Contagem de Trocas:** A questão 69 foca na **quantidade de trocas de posições** realizadas. Para determinar isso, é preciso simular o algoritmo passo a passo e contar cada vez que dois elementos adjacentes são efetivamente trocados de lugar.

- * **Exemplo de Passagem (para ordenação crescente):**

- * Vetor: `[A, B, C, D]`

- * Comparar A e B: Se $A > B$, troca.

- * Comparar B e C: Se $B > C$, troca.

* Comparar C e D: Se $C > D$, troca.

* Ao final de uma passagem, o maior elemento "borbulhou" para a última posição não ordenada.

• **Complexidade:** Embora simples, o Bubble Sort é ineficiente para grandes conjuntos de dados, com uma complexidade de tempo de $O(n^2)$ no pior e caso médio, onde 'n' é o número de elementos.

II. Estruturas de Dados Lineares: Fila (Queue)

Fila é uma estrutura de dados linear que segue o princípio **FIFO (First-In, First-Out)**, ou seja, o primeiro elemento a entrar é o primeiro a sair. Pense em uma fila de pessoas em um banco: a primeira pessoa a chegar é a primeira a ser atendida.

• Princípio FIFO (First-In, First-Out):

* **"Primeiro a Entrar, Primeiro a Sair"**: Isso significa que os elementos são adicionados em uma extremidade (o "final" ou "cauda" da fila) e removidos da outra extremidade (o "início" ou "cabeça" da fila).

• Operações Fundamentais:

1. ENFILEIRAR (Enqueue):

* **Função:** Adiciona um novo elemento ao **final (cauda)** da fila.

* **Exemplo:** `ENFILEIRAR(X)` adiciona `X` ao final da fila.

2. DESENFILAR (Dequeue):

* **Função:** Remove e retorna o elemento que está no **início (cabeça)** da fila.

* **Exemplo:** `DESENFILAR()` remove o elemento mais antigo da fila e o retorna. Se a fila estiver vazia, geralmente retorna um erro ou um valor nulo.

• **Representação do Estado da Fila:** A questão 70 especifica que a representação do estado da fila lista os elementos da esquerda para a direita, onde o **primeiro elemento (mais à esquerda) é o mais antigo** (o próximo a ser desenfileirado).

Para resolver a questão 70, é crucial simular cada operação de ENFILEIRAR e DESENFILAR, mantendo o controle de qual elemento está no início da fila e qual está no final, sempre respeitando o princípio FIFO.

Compreender esses conceitos e praticar a simulação de suas operações é a chave para resolver as questões de forma eficaz.

Questões de Provas Anteriores

Fonte: `escrituario_agente_de_tecnologia (1).pdf`, Página: 26

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3V
uLCAYNyBKdWwgMjAyNSAyMzo0Nzo0MCAtMDMwMA==
www.pciconcursos.com.br

26

BANCO DO BRASIL

AGENTE DE TECNOLOGIA - Microrregião 16 DF-TI GABARITO 1

69

As agências bancárias negociam seguros residenciais com seus clientes e, muitas vezes, precisam arquivar cópias de forma ordenada para que consultas eventuais sejam facilitadas. O gerente de uma agência precisava ordenar um vetor de documentos referentes a esses seguros, e o seu adjunto, da área de TI, o aconselhou a usar o algoritmo de ordenação chamado Bubble Sort.

Utilizando-se o algoritmo sugerido, qual será a quantidade de trocas de posições realizadas para ordenar, de modo crescente, o vetor de números de contrato (77, 51, 11, 37, 29, 13, 21)?

(A) 14

(B) 15

(C) 16

(D) 17

(E) 18

70

Uma das formas de o gerente de uma agência bancária acompanhar a qualidade dos serviços prestados aos seus clientes é verificar o estado da ordem de atendimento em vários instantes ao longo do expediente. O sistema que a gerência utiliza para tal fim é a estrutura de dados conhecida como FILA, que mostra a situação da ordem de atendimento no instante da verificação.

Nesse contexto, implementa-se uma estrutura de FILA de números inteiros com suas duas operações tradicionais:

ENFILEIRAR(Z), que ocorre no instante em que um cliente recebe uma senha Z e entra na FILA; e DESENFILEIRAR(),

que ocorre quando um cliente sai da FILA, caso em que DESENFILEIRAR() retorna o número da senha. Sabe-se, tam-

bém, que a representação do estado da FILA em um instante qualquer é realizada listando os elementos, de forma que o

primeiro elemento, da esquerda para a direita, é o mais antigo presente na FILA.

Nas condições apresentadas, considere uma FILA que começa vazia e realiza as seguintes operações:

ENFILEIRAR(8) → ENFILEIRAR(9) → DESENFILEIRAR() → ENFILEIRAR(10) → ENFILEIRAR(11) →

ENFILEIRAR(DESENFILEIRAR ()) → ENFILEIRAR(12) → DESENFILEIRAR() →
ENFILEIRAR(13) → DESENFILEIRAR()

Após realizar as operações acima, a FILA estará no estado

(A) 10 – 11 – 12

(B) 9 – 12 – 13

(C) 9 – 10 – 11

(D) 8 – 10 – 11

(E) 8 – 9 – 10

RASCUNHO