

### Ordenação Interna: Algoritmos de Troca

Considere a implementação em C da função classificação por trocas *BubbleSort* para resolver os exercícios a seguir.

```
int main()
{
    int vetor[] = {13, 11, 25, 10, 18, 21, 23};
    int tamanho=7, i, j, temp;

    for(i = tamanho - 1; i > 0; i--)
        for(j = 0; j < i; j++)
            if(vetor[j] > vetor[j+1]) /* compara elementos vizinhos */
            {
                int temp;

                temp = vetor[j];      /* troca vetor[j] e vetor[j+1] */
                vetor[j] = vetor[j+1];
                vetor[j+1] = temp;
            }
}
```

Realize a ordenação crescente do vetor a seguir utilizando o algoritmo *BubbleSort* e responda:

vetor[] = {13, 11, 25, 10, 18, 21, 23}

01. Em quantas varreduras o vetor é classificado? \_\_\_\_\_

02. Como identificar quantas chaves já estão classificadas?

03. Qual o melhor caso?

04. Qual o pior caso?

**05.** O algoritmo *BubbleSort* é **estável**? Lembrando que um algoritmo de ordenação estável é aquele que não altera a ordem relativa das chaves iguais.

(   ) Sim      (   ) Não

**06.** Invente um vetor de entrada que comprove a resposta da questão **(e)** e demonstre a execução passo a passo do algoritmo.

**07.** Podemos afirmar que a implementação do algoritmo *BubbleSort* é simples e a implementação é pequena?

(   ) Sim      (   ) Não

**08.** Quais as vantagens e desvantagens do algoritmo *BubbleSort*?

**09.** Suponha um vetor de comprimento N que possui valores que representam as cores Vermelha, Branca, Azul e Verde. Escreva um programa que ponha as cores neste vetor na seguinte ordem: Cores Vermelhas no extremo esquerdo do vetor, cores Azuis no extremo direito, cores Brancas após o conjunto de cores Vermelhas, e cores Verde após as cores Brancas e antes das cores Azuis. É possível utilizar o mesmo método do *BubbleSort* para resolver esse problema? Proponha uma solução na linguagem C.

Em seguida, responda:

- Qual a menor quantidade de passagens sobre o vetor necessária para a ordenação?

**Atenção: NÃO** é permitido contar as quantidades de cada cor e depois gerar o novo vetor

**10.** Especifique um algoritmo para avaliar o grau de desorganização das chaves incluídas em um vetor.