

**Estruturas de Dados e Algoritmos - EDA. Turma 1. Prof. Nilton Silva. 28/06/2023**  
**Correção da Prova 2**

**Questão 1.**

- a. FALSO.  $f_1(n)$  é menor (mais rápida) que  $f_2(n)$  para todo  $n < 649$ .
- b. **VERDADEIRO.  $f_2 = O(n^k)$  para qualquer  $k \geq 1$ ;**
- c. FALSO.  $f_1(n)$  é menor (mais rápido) que  $f_2(n)$  para todo  $n < 649$ .
- d. **VERDADEIRO.  $f_1 = \Omega(n^k)$  para qualquer  $k \leq 2$ ;**
- e. FALSO.  $f_2(n)$  só é mais rápida que  $f_1(n)$  quando  $n > 648$ .

**Questão 2.**

- a. ordena1: Bubble Sort.
- b. ordena2: Insert Sort.
- c. ordena3: Selection Sort

**Questão 3. V V F V**

**Questão 4.**

```
int buscaBinaria(int vetor[], int tamanho, int chave) {
    int inicio = 0, meio, fim = tamanho - 1;
    while (inicio <= fim) {
        meio = (fim+inicio)/2; //C1
        if (vetor[meio] == chave)
            return meio; //C2
        if (vetor[meio] < chave)
            inicio = meio + 1; //C3
        else
            fim = meio - 1; //C4
    }
    return -1;
}
```

- a. C1:  $\text{meio} = (\text{inicio} + \text{fim}) / 2$ ;
- b. C2: `return meio`;
- c. C3:  $\text{inicio} = \text{meio} + 1$
- d. C4:  $\text{fim} = \text{meio} - 1$

**Questão 5.**

```
int buscaSequencialOrdenada(int *V, int n, int chave) {
    int p = 0, posic = -1;
    while ( p < n && chave >= V[p] ) { //C1 && C2
        if (V[p] == chave) {
            posic = p; //C3
            break; //C4
        }
        p++;
    }
    printf("%d - %d\n", p, V[p]);
    return posic;
}
```

- a. C1:  $p < n$ ;
- b. C2:  $\text{chave} \geq V[p]$ ;
- c. C3:  $\text{posic} = p$ ;
- d. C4: `break`;

**Lembre-se:** Diferente de uma busca em vetor desordenado, na busca sequencial ordenada o laço deve restringir a busca levando em consideração que o vetor está ordenado. Ou seja, o laço de busca deve ser encerrado quando aparecer o primeiro valor no vetor maior que a chave.