Lista de Exercícios

Algoritmos de Busca e Comparação de Algoritmos

Exercício 01

Considere a implementação do algoritmo *Busca Sequencial* apresentada na aula. Faça uma versão deste algoritmo, trocando a estrutura de repetição *for* por *while*.

Exercício 02

Faça um programa que use o método de busca binária para encontrar e imprimir a posição de um determinado elemento no vetor. Se o elemento não existir, seu programa deve imprimir o valor -1.

Digite a quantidade de elementos do vetor: 5

Digite os elementos do vetor: 1 2 3 4 5

Digite o elemento procurado: 3

Posição do elemento procurado: 2

Digite a quantidade de elementos do vetor: 5

Digite os elementos do vetor: 4 8 7 2 1

Digite o elemento procurado: 3

Posição do elemento procurado: -1

Exercício 03

Altere o método de busca binária para que, a cada divisão do vetor, ele imprima quantos elementos foram descartados na busca.

Digite a quantidade de elementos do vetor: 5

Digite os elementos do vetor: 1 2 3 4 5

Digite o elemento procurado: 1

1ª iteração: 3

2ª iteração: 1

Exercício 04

Analise a função a seguir:

```
int descubra(int vet[], int n, int k){
    int x=0;
    int y=0;
    for(int i=0; i<n; i++){
        if(vet[i]<k){ //comparação 1
            x+=vet[i];
        }
        if(vet[i]>k){ //comparação 2
            y+=vet[i];
        }
    }
    return y-x;
}
```

- O que essa função retorna?
- Quantas vezes a comparação 1 é realizada?
- Quantas vezes a comparação 2 é realizada?
- Qual é o total de comparações?

Exercício 05

A função a seguir implementa a versão recursiva da busca binária. Complete os valores das lacunas.

```
bool buscaR(int v[], int n, int procurado, int esq, int dir){
    int meio = (esq+dir)/2;
    if(esq<=dir){
        if(v[meio] == procurado)
            return true;
        else if(v[meio] > procurado)
            return buscaR(v, n, procurado, _____, _____);
        else if(v[meio] < procurado)
            return buscaR(v, n, procurado, _____, _____);
    }
    return false;
}
```