

# Unidade I:

## Introdução - Algoritmos de Pesquisa

## Pesquisa sequencial:

1. exemplo:

```
boolean resp = false;

for (int i = 0; i < n; i++){
    if (array[i] == x){
        resp = true;
        i = n;
    }
}
```

a. Qual operação é a operação relevante?

**Resposta:**

Comparação entre elementos dos arrays.

b. Quantas vezes ele será executada?

**Resposta:**

Melhor caso – 1 vez

Pior caso – n vezes

## Exercício Resolvido (1):

Quando acontece o melhor e o pior caso do algoritmo de pesquisa sequencial?

**Resposta:**

Melhor caso – Quando o elemento a ser procurado está no primeiro index.

Pior caso – Quando o elemento a ser procurado está no ultimo index.

## Exercício Resolvido (2):

Supondo que temos uma informação extra sobre o array: que ele está ordenado. Conseguimos fazer algo mais eficiente?

**Resposta:**

Sim, podemos fazer uma pesquisa binaria, ao invés de sequencial.

## Pesquisa sequencial:

### 1. Exemplo:

```
boolean resp = false;
int dir = n - 1, esq = 0, meio;
while (esq <= dir) {
    meio = (esq + dir) / 2;
    if (x == array[meio]){
        resp = true;
        esq = n;
    } else if (x > array[meio]){
        esq = meio + 1;
    } else {
        dir = meio - 1;
    }
}
```

- a. Qual é a operação relevante?

**Resposta:**

As comparações entre os elementos da array

- b. Quantas vezes ela será executada?

**Resposta:**

Melhor caso – 1 vez

Pior caso –  $\lg(n)$

## Exercício Resolvido (3):

Quando acontece o melhor e o pior caso do algoritmo de pesquisa binária?

**Resposta:**

Melhor caso, ele acontece quando o elemento a ser procurado está no meio do array, ou seja, a primeira posição a ser procurada.

Pior caso, ele acontece quando o elemento está na última posição procurada, ou quando não está no array.

## Exercício (1):

A solução apresentada para pesquisa binária faz duas comparações entre elementos do array em cada execução da repetição, modifique o código para efetuarmos apenas uma

Resposta:

```
static boolean findInt(int[] array, int find){
    boolean resp = false;
    int n = array.length;
    int dir = n - 1, esq = 0, meio = 0;

    meio = (esq + dir)/2;
    while(esq < dir){
        if(find > array[meio]){
            esq = meio + 1;
        }
        else{
            dir = meio;
        }
        meio = (esq + dir)/2;
    }

    if(find == array[meio]){
        resp = true;
    }

    return resp;
}
```

Obs:

Arquivo encontra-se dentro da pasta ***exercicios\_praticos***.

O enunciado diz que a cada repetição do while, são feitas duas comparações, este código faz uma comparação para cada repetição, porém ainda sim faz duas comparações no método.

Ao meu ver não é possível fazer este programa ter somente uma comparação no método.

Por favor desconsidere este comentário caso tenha interpretado errado o enunciado.