

# Algoritmos e Programação

## Métodos de Pesquisa

Prof. Dr. Joaquim Assunção

CENTRO DE TECNOLOGIA  
UFSM  
2025



# Dois métodos principais

- Linear
- Binária

# Linear (ou Sequencial)

- Busca a informação desejada a partir de um elemento inicial, até chegar a um elemento final.
- A ordem não importa, o elemento inicial pode ser o último do vetor; mas neste caso, o final será o primeiro elemento do vetor.

# Pesquisa Linear (ou Sequencial)

- Exemplo.
- Considerando os vetores abaixo. Retorne a posição do valor 9.

1	3	6	7	9	20	30	40	50
---	---	---	---	---	----	----	----	----

9	10	15	16	19	20	30	40	50
---	----	----	----	----	----	----	----	----

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

# Pesquisa Binária

- É mais rápido que o método de pesquisa sequencial, porém exige que o vetor esteja previamente ordenado.
- Divide o vetor em duas partes (binária) e procura saber se a informação está na posição central após a divisão.
  - Se estiver na posição central, a informação é apresentada
  - Se não, verifica se a informação está na primeira ou segunda parte do vetor (desprezando a outra parte)

# Pesquisa Binária

- Exemplo.
- Considerando os vetores abaixo. Retorne a posição do valor 9.

1	3	6	7	9	20	30	40	50
---	---	---	---	---	----	----	----	----

9	10	15	16	19	20	30	40	50
---	----	----	----	----	----	----	----	----

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

# Pesquisa Binária

```
Inteiro: inicio, fim, meio, posicao, pesq;  
Lógico: achou;
```

```
achou <- falso;  
inicio <- 0; meio <- 0; fim <- tam -1;  
  
enquanto(inicio <= fim && achou = falso) {  
    meio <- (inicio + final) div 2  
    se (pesq == vetor[meio]) {  
        achou <- verdadeiro;  
        posicao <- meio + 1;  
    } senão se (pesq < vetor[meio]) {  
        fim <- meio - 1;  
    } senão {  
        inicio <- meio + 1;  
    }  
}
```

# Pesquisa Binária

Inteiro: inicio, fim, meio, posicao, pesq;  
Lógico: achou;

```
achou <- falso;
inicio <- 0; meio <- 0; fim <- tam -1;

enquanto(inicio <= fim && achou = falso) {
  meio <- (inicio + final) div 2
  se (pesq == vetor[meio]) {
    achou <- verdadeiro;
    posicao <- meio + 1;
  } senão se (pesq < vetor[meio]) {
    fim <- meio - 1;
  } senão {
    inicio <- meio + 1;
  }
}
```

```
int buscaBinaria (int arr[], int size, int target) {
  int left = 0;
  int right = size - 1;

  while (left <= right) {
    int mid = left + (right - left) / 2;

    if (arr[mid] == target)
      return mid;
    else if (arr[mid] < target)
      left = mid + 1;
    else
      right = mid - 1;
  }

  return -1; // não encontrou
}
```

# Exercícios

1. Crie um algoritmo sequencial que recebe um vetor de notas e faz a busca por uma determinada nota. O algoritmo deve retornar o índice da nota.
2. Crie uma função de pesquisa binária para resolver o problema 1.