



Algoritmos e Estruturas de Dados I

Prof^a Priscilla Abreu
priscilla.abreu@ime.uerj.br
2022.1

Algoritmos e Estruturas de Dados I



Roteiro da aula

- Listas: introdução
- Busca e ordenação:
 - Operação de busca

Algoritmos e Estruturas de Dados I



Disponibilização de materiais e avisos:

- Google Classroom:

vurjz3i



<https://classroom.google.com/c/NDg4NDE4MDk5NTA4?cjc=vurjz3i>



Listas Introdução

Algoritmos e Estruturas de Dados I



Listas

O QUE É UMA LISTA?

Consideraremos como listas conjuntos sem repetições!

Uma lista é um conjunto de dados relacionados, e de número variável de elementos.

- Exemplo:
 - Lista de alunos de uma turma;
 - Lista de aprovados em um concurso;
 - Lista de produtos de uma loja;
 - ...

Cada elemento da lista terá uma chave – identificador único.

Algoritmos e Estruturas de Dados I



Operações comuns:

- **Busca;**
- **Inserção;**
- **Remoção.**

Outras operações:

- Alteração de um elemento da lista;
- Ordenação dos elementos da lista segundo uma determinada chave;
- Determinação do primeiro elemento da lista, ...

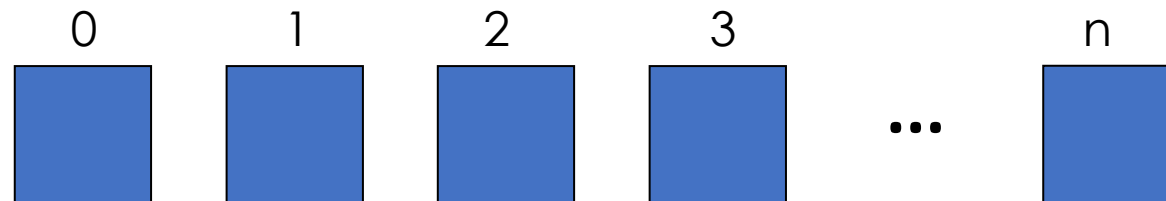
Algoritmos e Estruturas de Dados I



Listas Linear

Estrutura que permite representar um conjunto de dados de forma a preservar a relação de ordem existente entre eles.

Temos um primeiro elemento, segundo elemento, ..., n-ésimo elemento.



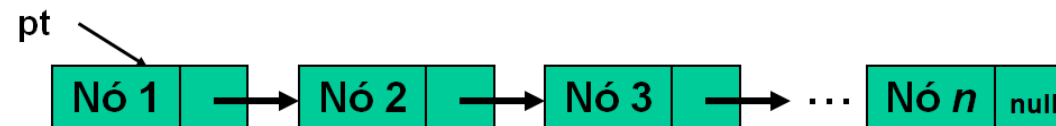
Algoritmos e Estruturas de Dados I



O tipo de armazenamento de uma lista linear pode ser classificado de acordo com a posição relativa na memória (contínua ou não) de cada dois nós consecutivos na lista.

Existem dois tipos de listas:

- Lista sequencial
- Lista encadeada





BUSCA EM ARRAYS

Algoritmos e Estruturas de Dados I



Pense em sistemas que costumamos acessar, tais como como sistemas de compra (online ou lojas físicas), sistemas de agendamento de serviços, Redes Sociais, entre outros.

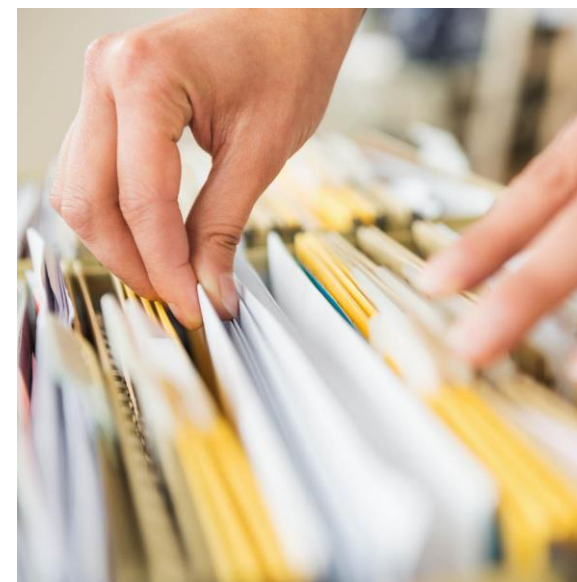
Quais são as principais operações em sistemas que geralmente utilizamos?

Algoritmos e Estruturas de Dados I



Introdução

Uma importante operação é a **busca**. Todos fazemos buscas, consciente ou inconscientemente, ao longo do dia.



Algoritmos e Estruturas de Dados I



Introdução

Grande parte dos sistemas computacionais utilizam operações de busca.

The screenshot shows the Saraiva website interface. The search bar contains 'estrutura de dados'. Below the search bar, there are several suggested products:

- Estrutura de Dados e Algoritmos Em C++ - Tradução da 4ª... Por: R\$ 150,9
- Estrutura de Dados Descomplicada - Em Linguagem C Por: R\$ 133,9
- Estrutura De Dados E Algoritmos Usando C - Fundamentos ... Por: R\$ 109
- Estruturas de Dados e seus Algoritmos - 3ª Ed. 2010 Por: R\$ 170
- Programação - Algoritmos e Estrutura de Dados - 3ª Ed. ... Por: R\$ 123

On the right side of the page, there is a promotional banner for 'DUMBO' with the text 'A PARTIR DE R\$ 34,90'.

The screenshot shows a 'Consulta Produtos' window. It contains a table with the following data:

Código	Descrição	Referência	Und	Custo	Marca
603	BOBINA BFF 0,90MM	BFF90	KG	.	BFF 0,90MM
657	BOBINA BFQ 1,50MM	BFQ150	KG	.	BFQ 1,50MM
605	BOBINA BFQ 1,80MM	BFQ180	KG	.	BFQ 1,80MM
651	BOBINA BFQ 2,00MM	BFQ200	KG	.	BFQ 2,00MM
658	BOBINA BFQ 2,16MM	BFQ216	KG	.	BFQ 2,16MM
652	BOBINA BFQ 2,25MM	BFQ225	KG	.	BFQ 2,25MM
650	BOBINA BFQ 2,40MM	BFQ240	KG	.	BFQ 2,40MM
649	BOBINA BFQ 2,50MM	BFQ250	KG	.	BFQ 2,50MM

Below the table, there are two sections: 'Estoque' (Inventory) and 'Preços' (Prices).

Estoque:

Local	Descrição	Sub-Item	Estoque Físico
1	ESTOQUE 1	0	0,000
7	ESTOQUE 7	0	40.105,000

Preços:

Tab	Descrição	Preço + IPI
6	TABELA 6	1,00
7	TABELA 7	0,00

At the bottom right, there is a 'Total' value of 40.105,000.



Uma situação mais simples...

Como efetuar uma busca em um conjunto de valores, tais como em vetores?

Algoritmos e Estruturas de Dados I



Considere um vetor com 10 números inteiros.

4	5	98	32	41	22	7	65	18	57
---	---	----	----	----	----	---	----	----	----

Suponha que você queira verificar se o número 41 está no vetor. Como seria feita essa verificação?

Como seria um algoritmo para representar esse processo?

Algoritmos e Estruturas de Dados I



Dada uma coleção de n elementos, pretende-se saber se um determinado elemento/valor está presente nesta coleção.

Neste curso, vamos considerar que essa coleção de elementos é implementada como um vetor de n elementos: $vet[0] \dots vet[n-1]$.

Algoritmos e Estruturas de Dados I




- Busca Sequencial ou linear
 - Não ordenada
 - Ordenada
- Busca Binária

Algoritmos e Estruturas de Dados I



Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----


A blue arrow points upwards from below the first cell of the array, which contains the value 14.

Algoritmos e Estruturas de Dados I



Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----


A blue arrow points upwards from below the second cell of the array, which contains the value 5.

Algoritmos e Estruturas de Dados I



Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----


A blue arrow points upwards from below the array to the element 21, which is at index 2 (0-based indexing).

Algoritmos e Estruturas de Dados I



Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----




Algoritmos e Estruturas de Dados I



Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----


A blue arrow points upwards from below the array, specifically pointing to the element '8' in the fifth position of the array.

Algoritmos e Estruturas de Dados I



Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----

A blue arrow points upwards from below the array to the element 49, which is the sixth element in the sequence.

Algoritmos e Estruturas de Dados I



Busca Sequencial

**TENTE
IMPLEMENTAR!!!**

- Método bastante simples;
- Utilizado quando os dados não estão ordenados;
- Percorre o vetor, elemento por elemento, verificando se o elemento desejado está presente no vetor e termina:
 - **Com sucesso:** valor pesquisado é encontrado;
 - **Sem sucesso:** todos os registros são pesquisados, mas o valor não é encontrado.

Algoritmos e Estruturas de Dados I



Busca Sequencial

```
int buscaSeq(int n, int valor) {  
    int i;  
    for (i=0; i<n; i++){  
        if (vet[i] == valor)  
            return i;  
    }  
    return -1;  
}
```


Algoritmos e Estruturas de Dados I



Busca Sequencial

Tratamento da Informação:

```
pos= buscaSeq(n, valor);  
if (pos !=-1)  
    printf(" %d foi encontrado na posição %d", valor, pos);  
else  
    printf(" %d não foi encontrado na lista.", valor);
```

Algoritmos e Estruturas de Dados I



Busca Sequencial

Quantas vezes a comparação `valor == vetor[i]` é executada?

Caso o elemento não esteja no vetor, n vezes;

Caso o elemento esteja no vetor:

- 1 vez, no melhor caso;
- N vezes, no pior caso;
- $N/2$ vezes, no caso médio;

Complexidade
 $O(n)$

Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Considere que o vetor a ser utilizado para a busca esteja ordenado de modo crescente.

Como poderíamos realizar a busca de forma mais eficiente?

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----


Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Considere que estamos procurando pelo valor **40**.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----




Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Considere que estamos procurando pelo valor **40**.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----




Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Considere que estamos procurando pelo valor **40**.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----




Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Considere que estamos procurando pelo valor **40**.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----



Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Considere que estamos procurando pelo valor **40**.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----



48 <= 40 ?

Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Logo, quando o valor em uma determinada posição ultrapassar o valor procurado, não precisamos continuar a busca.

Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

Logo, quando o valor em uma determinada posição ultrapassar o valor procurado, não precisamos continuar a busca.

**TENTE
IMPLEMENTAR!!!**

Algoritmos e Estruturas de Dados I



Busca Sequencial Ordenada

```
int buscaOrd(int n, int valor) {  
    int i;  
    for (i=0; i<n; i++){  
        if (vet[i] == valor)  
            return i;  
        if (vet[i] > valor)  
            return -1;  
    }  
    return -1;  
}
```

Algoritmos e Estruturas de Dados I



Busca Binária

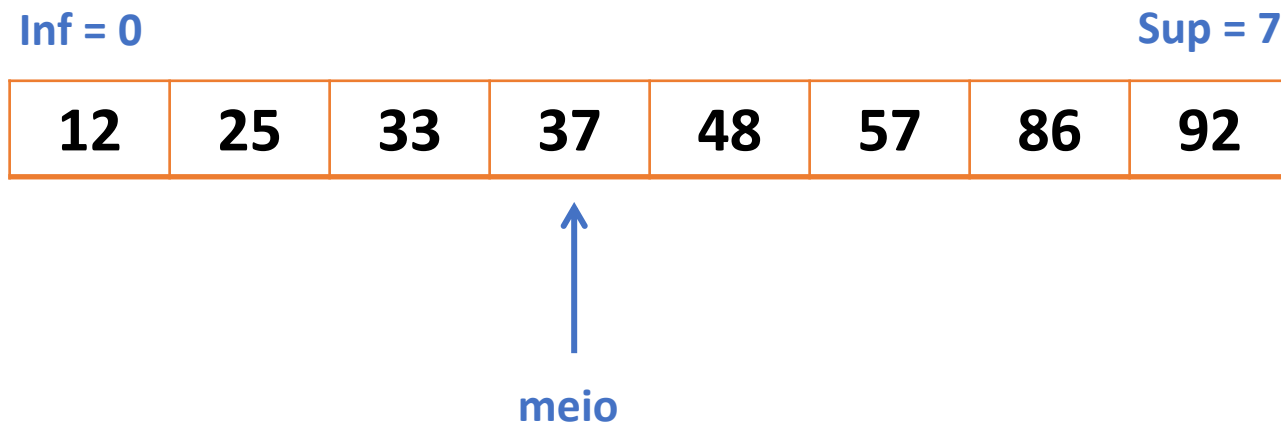
Considerando um vetor ordenado de forma crescente podemos aplicar outro método de busca: a busca binária.

Algoritmos e Estruturas de Dados I



Busca Binária

Exemplo: Busca pelo valor 86.

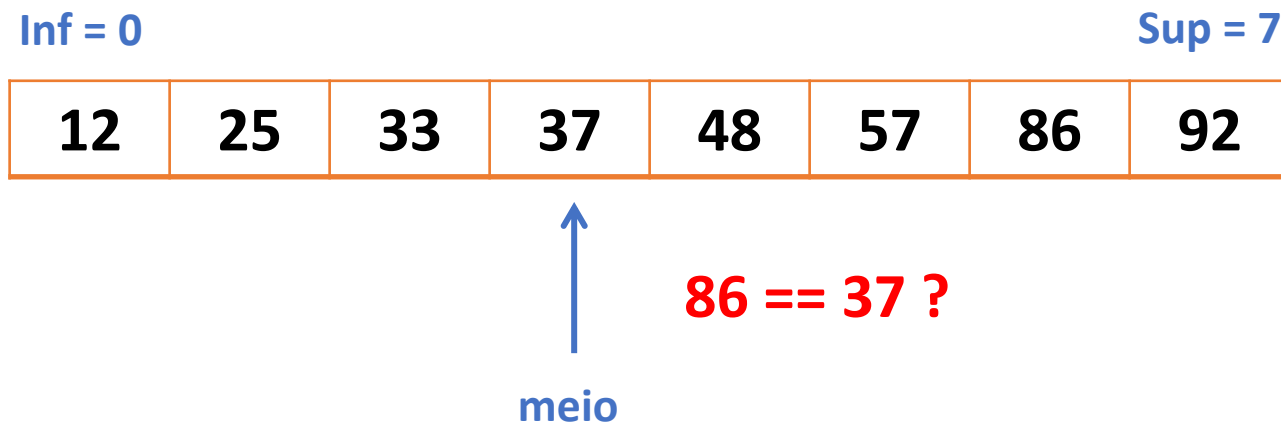


Algoritmos e Estruturas de Dados I



Busca Binária

Exemplo: Busca pelo valor 86.

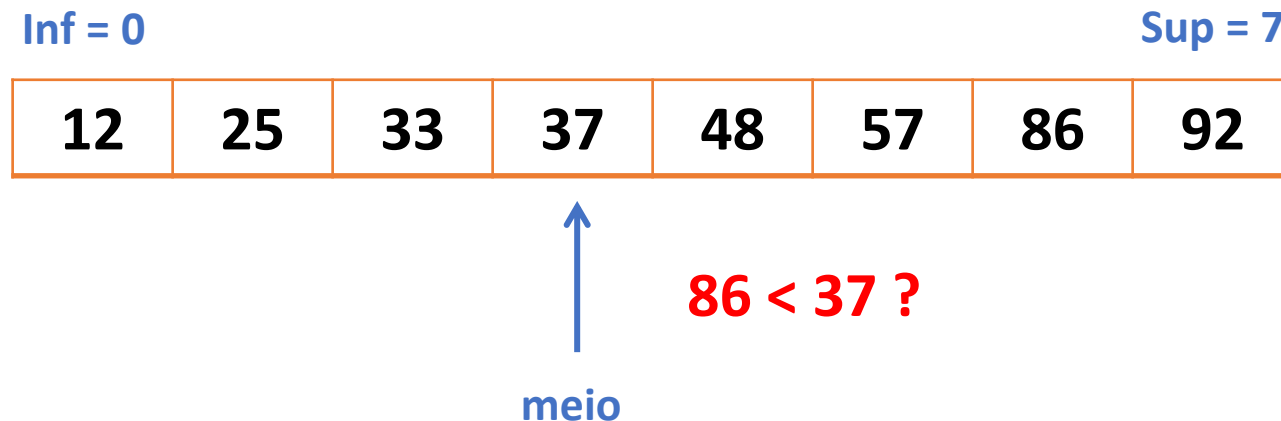


Algoritmos e Estruturas de Dados I



Busca Binária

Exemplo: Busca pelo valor 86.

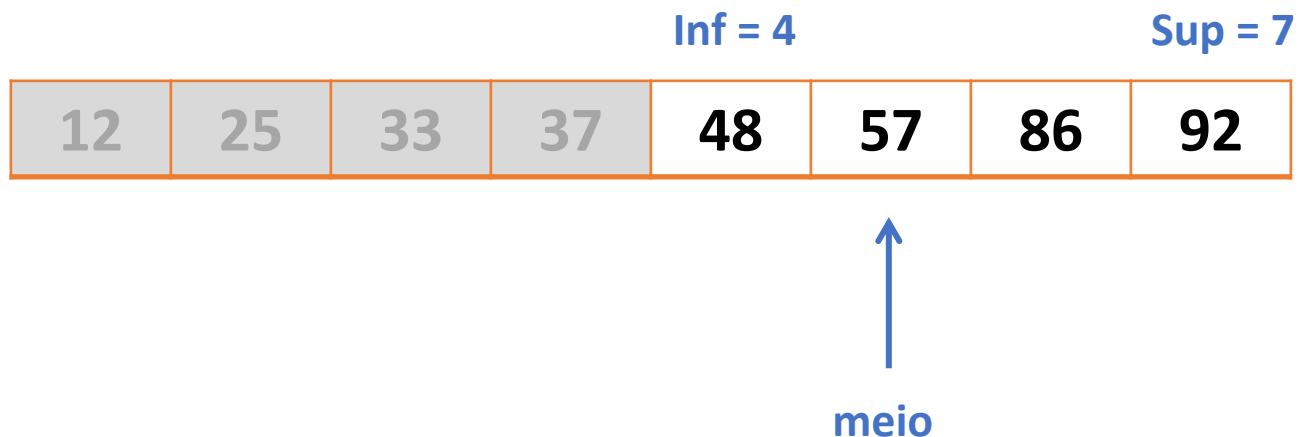


Algoritmos e Estruturas de Dados I



Busca Binária

Exemplo: Busca pelo valor 86.

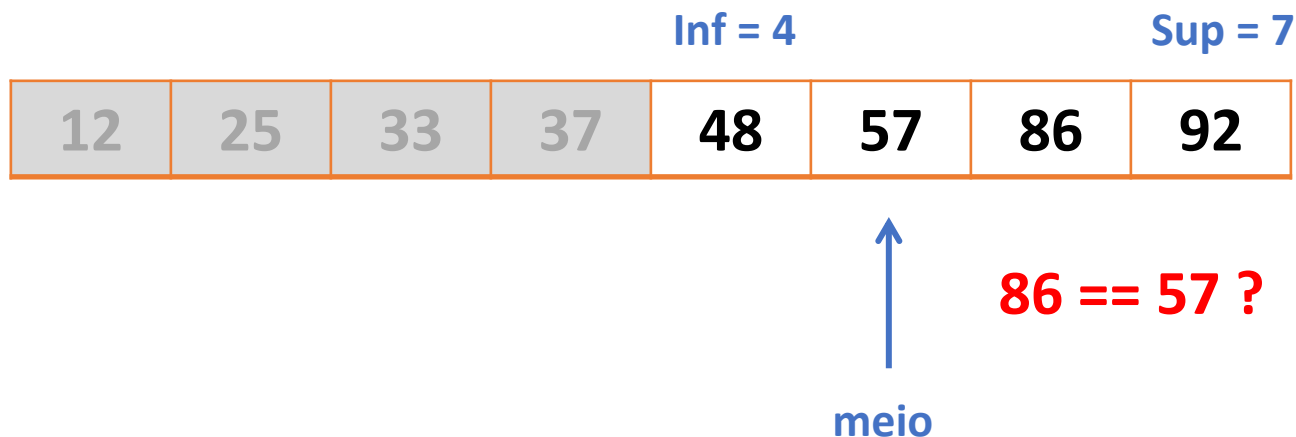


Algoritmos e Estruturas de Dados I



Busca Binária

Exemplo: Busca pelo valor 86.

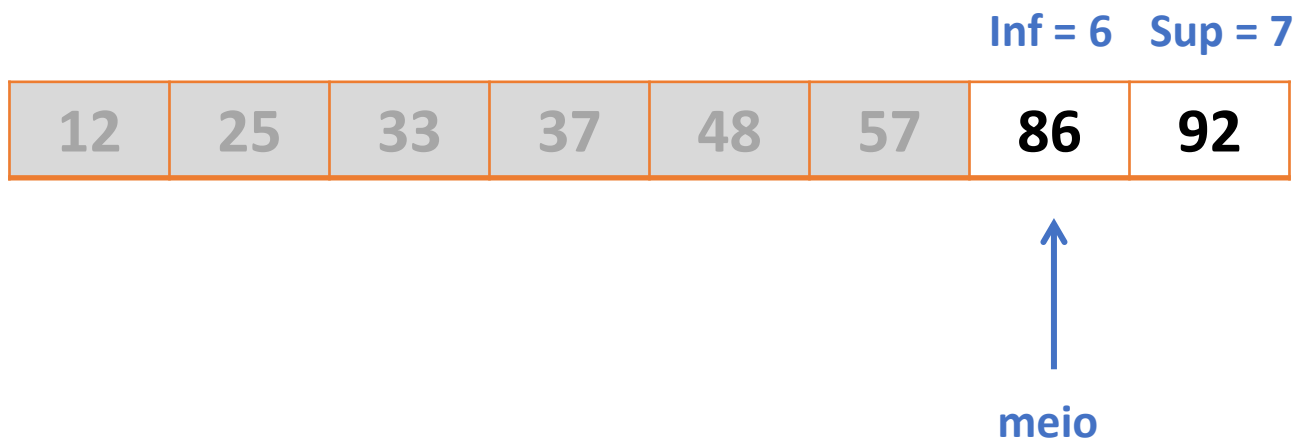


Algoritmos e Estruturas de Dados I



Busca Binária

Exemplo: Busca pelo valor 86.

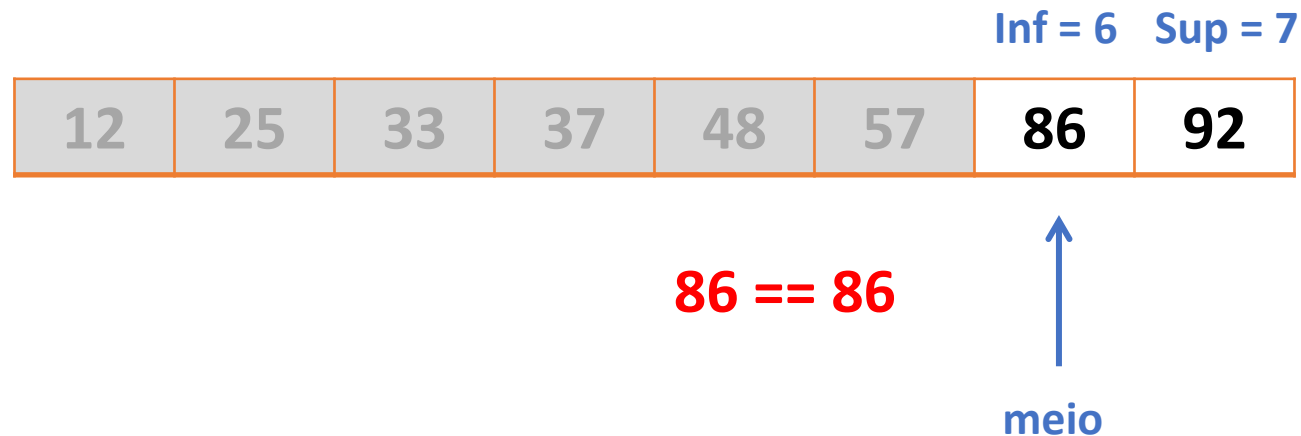


Algoritmos e Estruturas de Dados I



Busca Binária

Exemplo: Busca pelo valor 86.



Algoritmos e Estruturas de Dados I



Busca Binária

Complexidade
 $O(\log n)$

Em cada passo, o tamanho do vetor é reduzido pela metade.

Ao comparar um determinado elemento do vetor com o valor desejado é possível identificar:

- Se o elemento procurado é o elemento comparado;

- Se o elemento procurado está antes ou depois do elemento comparado.

Se após diversas divisões, a lista possuir tamanho 0 (zero), então o elemento não está na lista.

Algoritmos e Estruturas de Dados I



Busca Binária

O elemento buscado é comparado ao elemento do meio da lista:

- Se igual, busca bem-sucedida;
- Se menor, busca-se na metade inferior da lista;
- Se maior, busca-se na metade superior da lista.

Algoritmos e Estruturas de Dados I



Busca Binária

```
int pesqBin (int valor, int n){  
    int inicio, fim, meio;  
    inicio= 0; fim=n-1;
```

```
        while (inicio <= fim) {  
            meio = floor( (inicio + fim) / 2 );  
            if (valor == v[meio])  
                return(meio);  
            else  
                if (valor > v[meio])  
                    inicio= meio + 1  
                else  
                    fim= meio -1;  
        }  
        if (inicio>fim) return -1;  
    }
```

Algoritmos e Estruturas de Dados I



Busca Binária

Complemente sua implementação, acrescentando um menu de opções para que, após informar um conjunto de valores, o usuário possa escolher que tipo de busca efetuar.

Exemplo:

Escolha sua busca

- 1- Busca sequencial
- 2- Busca sequencial ordenada
- 3- Busca binária

Lembre-se:

As buscas 2 e 3 só podem ser realizadas se o vetor estiver ordenado!

Sugestão:

Acrescente sua solução do exercício da semana passada, verificando se o vetor está ordenado!

Algoritmos e Estruturas de Dados I



Solução

```
#include <stdio.h>

int buscaSequencial(int vetor[], int chave, int tam){
    int cont;
    for(cont=0;cont<tam;cont++){
        if(vetor[cont] == chave)
            return cont;
    }
    return -1;
}
```

```
int buscaSeqOrd(int vetor[], int chave, int tam){
    int cont;
    for(cont=0;cont<tam;cont++){
        if(vetor[cont] == chave)
            return cont;
        if(vetor[cont] > chave)
            return -1;
    }
    return -1;
}
```

Algoritmos e Estruturas de Dados I



Solução

```
int buscaBinaria(int vetor[], int chave, int tam){
    int piso, teto, meio;
    piso = 0;
    teto = tam-1;
    meio = 0;
    while (piso <= teto){
        meio = (piso+teto)/2;
        if (vetor[meio] == chave)
            return meio;
        if(chave > vetor[meio])
            piso = meio +1;
        else
            teto = meio-1;
    }
    return -1;
}
```

```
int main() {
    int vet[10];
    int cont, chave, res;
    for(cont=0; cont <10; cont++){
        printf("Informe o valor da posição %d:",cont+1);
        scanf("%d",&vet[cont]);
    }
    printf("Que valor você deseja buscar? ");
    scanf("%d",&chave);
    //res = buscaSequencial(vet,chave,10);
    res = buscaSeqOrd(vet,chave,10);
    if(res!=-1)
        printf("\nO valor foi encontrado na posição: %d",res+1);
    else
        printf("\nValor não foi encontrado!");
    return 0;
}
```



FIM