

Busca Sequencial e Binária

Componentes:
Arthur Ramos
Leonardo Castro

Busca Sequencial

Definição: A busca sequencial compara cada posição da estrutura em busca do elemento requisitado onde ele ,para caso atinja ao final da array ou encontre o elemento procurado .

Modo de Funcionamento: A função da busca recebe um array a quantidade n de elementos desse array e o elemento que deseja encontrar. Então vai comparar o elemento com cada posição da lista e caso seja igual ira retornar ele ,caso atinja ao final da lista o algoritmo vai retornar que não encontrou o elemento .

5	4	3	2	1
---	---	---	---	---

5	4	3	2	1
---	---	---	---	---

5	4	3	2	1
---	---	---	---	---

5	4	3	2	1
---	---	---	---	---



Exemplo de busca :

```
int busca (int n, int*vet , int elem ){
    //recebe n : o tamanho do array ,
    //vet : o array e
    //elem: o elemento que quer encontrar
    for (int i = 0 ; i < n ; i++){//função para percorrer o vetor
        // percorre elemento por elemento do array até encontrar o que foi pedido
        if(elem == vet[i]){
            return i ;
            //Retorna o valor caso encontrado
        }
    }
    //se nao encontrar o elemento :
    return -1 ;
}
```



Vantagens e Desvantagens:

Vantagens

- Não necessita de ordenação prévia para funcionar
- Simples para implementar
- Eficiente para listas pequenas

Desvantagens

- É ineficiente para conjuntos de arrays muito grandes
- No pior caso percorre todos os elementos da lista



Busca Binária

Definição: A Busca Binária é um algoritmo eficiente para encontrar um elemento em um array ordenado. Diferente da busca linear, que examina cada elemento sequencialmente, a busca binária reduz pela metade o espaço de busca a cada iteração, tornando-a muito mais eficiente para grandes conjuntos de dados.

Funcionamento: O funcionamento da busca binária baseia-se na divisão do array em duas metades e na comparação do elemento do meio com o valor desejado. Se o valor buscado for menor que o elemento do meio, a busca continua na metade inferior do array, se for maior, continua na metade superior. Esse processo se repete até que o elemento seja encontrado ou até que o subarray tenha tamanho zero, indicando que o elemento não está presente.



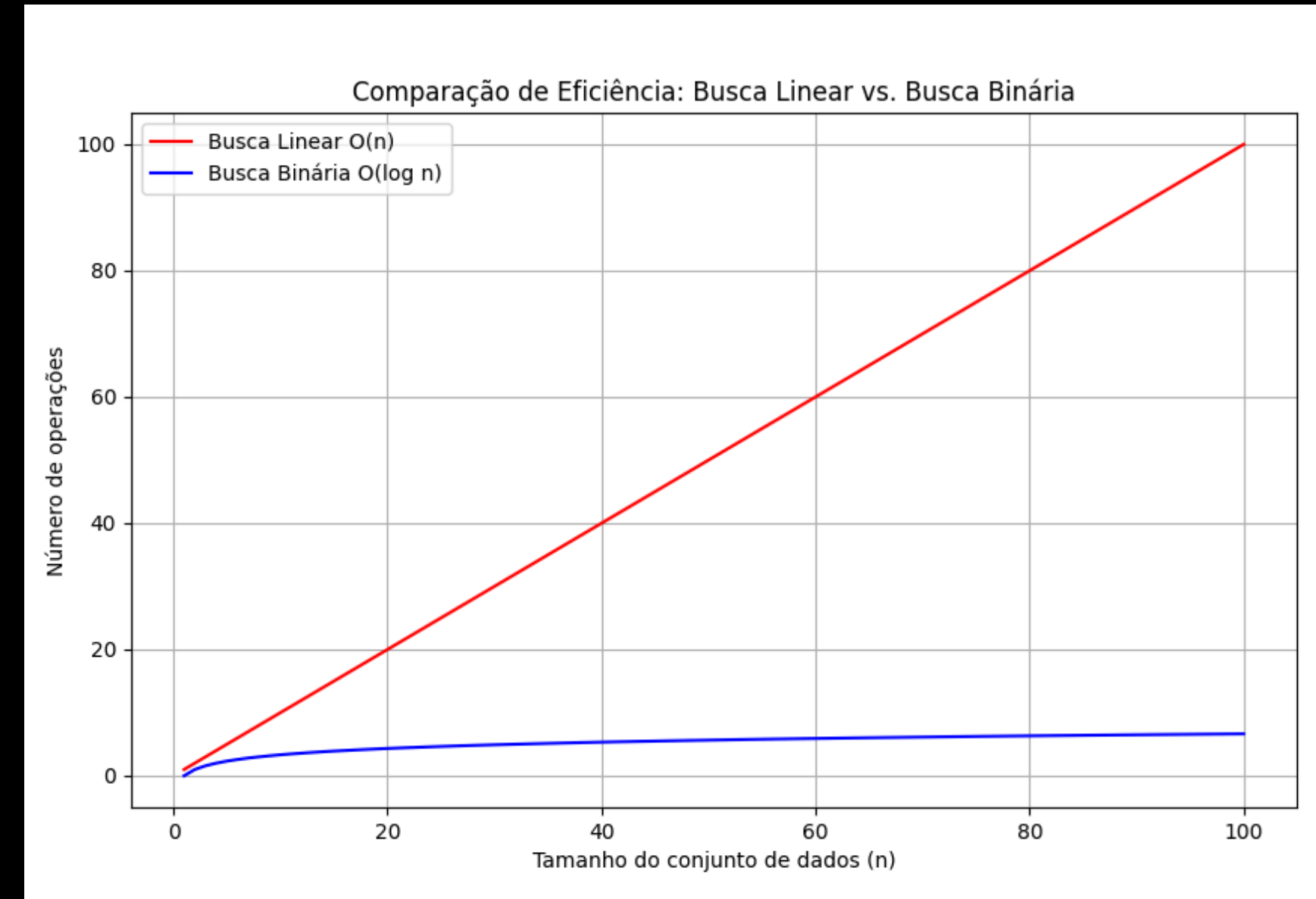
Algoritmo

```
1  #include <stdio.h>
2
3  // Função de Busca Binária
4  int busca_binaria(int n, int* vet, int elem){
5      int ini = 0;
6      int fim = n - 1;
7      int meio;
8
9      while (ini <= fim){
10         meio = (ini + fim) / 2;
11         if (elem < vet[meio]) // Ajusta o novo final
12             fim = meio - 1;
13         else if (elem > vet[meio]) // Ajusta o novo início
14             ini = meio + 1;
15         else
16             return meio; // Encontra o elemento
17     }
18     return -1; // Caso não encontre
19 }
```



Vantagens

- **Eficiência:** A busca binária tem complexidade $O(\log n)$, o que a torna muito mais rápida do que a busca linear $O(n)$ para grandes conjuntos de dados.
- **Redução de Comparações:** O algoritmo reduz o número de comparações em cada iteração, tornando a busca mais eficiente.
- **Facilidade de Implementação:** O algoritmo é simples de implementar e entender.



Desvantagens

- **Requer Ordenação Prévia:** Para que a busca binária funcione corretamente, o array precisa estar ordenado, o que pode adicionar um custo extra se a ordenação ainda não tiver sido feita.
- **Dificuldade de Uso com Listas Dinâmicas:** Em estruturas dinâmicas como listas encadeadas, a busca binária perde eficiência, pois o acesso a elementos no meio da lista é mais lento.



Referências

- **Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introdução a Algoritmos (4ª ed.). MIT Press.**
- **Celes, W., Cerqueira, R., & Rangel, J. L. (2004). Introdução a Estruturas de Dados com Técnicas de Programação em C (4ª tiragem). Elsevier.**
- **Bhargava, A. Y. (2017). Entendendo Algoritmos: Um Guia Ilustrado Para Programadores e Outros Curiosos. Novatec Editora.**