



Universidade Federal do Pará

Campus Universitário de Castanhal
Faculdade de Computação

Estruturas de Dados

Métodos de Pesquisa e
Ordenação ou classificação

Profª Penha Abi Harb
mpenha@ufpa.br



Métodos de Pesquisa Sequencial ou Linear

- Percorre o vetor desde a primeira posição até a última.
- Para cada posição i , comparamos $a[i]$ com x .
- Se forem iguais dizemos que x existe.
- Se chegarmos ao fim do vetor dizemos que x não existe.



Métodos de Pesquisa

- Esta operação consiste na verificação da existência de um determinado valor dentro de um conjunto de dados (um vetor ou um arquivo) e, em muitos casos, da posição do(s) elemento(s) com aquele valor.
- Veremos dois métodos de pesquisa bastante difundidos: Pesquisa Sequencial e Pesquisa Binária.



Código

```
public int procura(int[] vetor, int  
    elementoProcurado) {  
    for (int i = 0; i < vetor.length; i++) {  
        if (vetor[i] == elementoProcurado)  
            return i;  
    }  
    return -1; // Não achou, retorna -1  
}
```

G	J	E	H	F	C	D	A	I	B
---	---	---	---	---	---	---	---	---	---



Perguntas

- Quanto tempo é que este algoritmo demora para executar?
- Ou seja, quantas vezes é que a comparação $x == a[i]$ é executada?
- Sempre será eficiente?

G	J	E	H	F	C	D	A	I	B
---	---	---	---	---	---	---	---	---	---



Métodos de Pesquisa Binária

- Vetor previamente ordenados
- Princípio de reduzir à metade, sucessivamente, o “universo de busca” (paradigma de divisão e conquista)
- Determinar o elemento que está no meio do vetor e compará-lo com o valor procurado (chave).



Análise de Complexidade

- No melhor caso, o elemento a ser buscado é encontrado logo na primeira tentativa da busca.
- No pior caso, o elemento a ser buscado encontra-se na última posição e são feitas N comparações, sendo N o número total de elementos.
- No caso médio, o elemento é encontrado após $(N+1)/2$ comparações.
- O algoritmo de busca linear é um algoritmo Ordem de complexidade : $O(n)$.



Métodos de Pesquisa Binária

- Se o elemento central for igual à chave, a pesquisa termina;
- Se o elemento central for menor que a chave, a pesquisa continuará na metade superior (a inferior será desprezada);
- Já se o elemento central for maior que a chave, continua-se a pesquisa somente na metade inferior do vetor;
- E assim sucessivamente...



```
typedef float TDado;    // ou outro tipo qualquer

int BuscaBin (tvect vet, TDado valproc, int n)
{
    int baixo = 0;
    int alto = n-1;
    int meio;

    while (baixo <= alto)
    {
        meio = (baixo+alto) / 2;
        if (valproc < vet[meio] )
            alto = meio-1;
        else if (valproc > vet[meio] )
            baixo = meio+1;
        else
            return meio;
    }
    return -1;
}
```

10	20	30	40	50	60	70	80	90
0	1	2	3	4	5	6	7	8

valproc

n

baixo

alto

meio



Análise de Complexidade

- Podemos dizer que o melhor algoritmo para resolver um problema é aquele que possui a menor complexidade de tempo e espaço.
- Em outras palavras, é o algoritmo que, conforme a entrada cresce tendendo ao infinito, é aquele que apresenta a menor variação de tempo e memória utilizada para terminar.



Análise de Complexidade

- A eficiência da busca binária é muito superior a uma busca linear (no pior caso), pois a cada iteração metade do sub-vetor é descartada. Sua complexidade é de $O(\log(n))$ enquanto a complexidade da busca linear é $O(n)$.
- A cada teste, descartamos uma das metades do (sub)vetor pesquisado.



Análise de Complexidade

- Tipos:**
- Espacial: Este tipo de complexidade representa, por exemplo, o espaço de memória usado para executar o algoritmo.
- Temporal: Este tipo de complexidade é o mais usado podendo dividir-se em dois grupos:
 - Tempo (real) necessário à execução do algoritmo.
 - Número de instruções necessárias à execução.



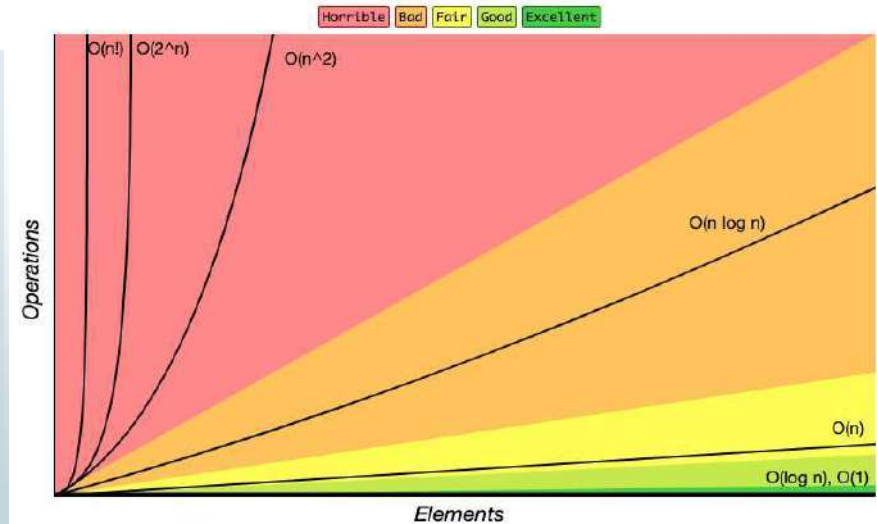
Análise de Complexidade

- A ideia da notação Big-O é descrever o comportamento geral do algoritmo em termos do crescimento do número de operações conforme cresce o número de elementos processados (a quantidade de itens é descrita, genericamente, por n).
- A ideia é usar a letra O seguida de uma função sobre n que descreva esse crescimento do algoritmo.



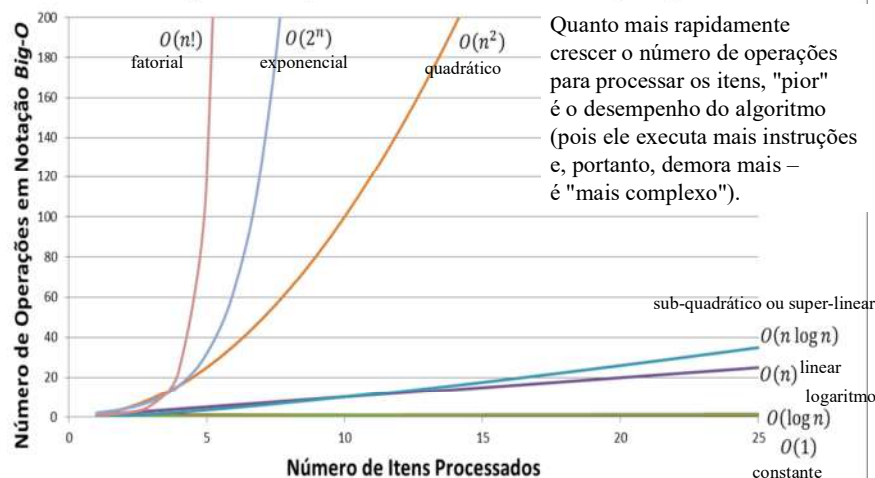
Análise de Complexidade

Big-O Complexity Chart



Análise de Complexidade

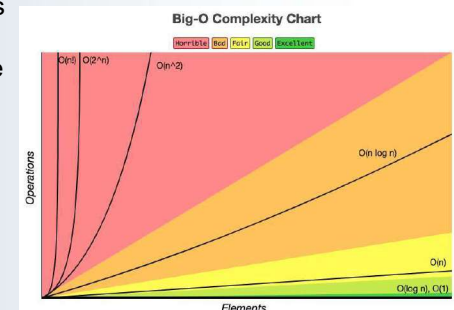
Ilustração das Complexidades Mais Comuns - Notação Big-O



Análise de Complexidade

- $O(n!)$ (fatorial) - pior comportamento para um algoritmo, pois rapidamente o processamento se torna inviável. (algoritmo que gere todas as possíveis permutações de uma lista)
- $O(2^n)$ (exponencial) também é bem ruim, pois o número de instruções também cresce muito rapidamente (exponencialmente). (busca em árvores binárias não ordenadas)
- $O(n^2)$ (quadrático) é factível, mas tende a se tornar muito ruim quando a quantidade de dados é suficientemente grande. (matriz bidimensional, dois for)

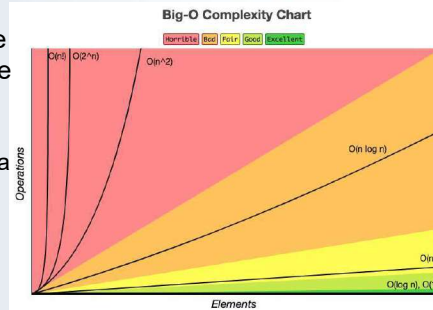
$O(n \log n)$ consegue otimizar algoritmos que são quadráticos em sua implementação mais direta. (QuickSort)





Análise de Complexidade

- $O(n)$ (linear) é aquele cujo crescimento no número de operações é diretamente proporcional ao crescimento do número de itens. (algoritmos de busca em uma matriz unidimensional não ordenada)
- $O(\log n)$ (logaritmo) é aquele cujo crescimento do número de operações é menor do que o do número de itens. (algoritmos de busca em árvores binárias ordenadas).
- $O(1)$ (constante) é aquele em que não há crescimento do número de operações, pois não depende do volume de dados de entrada (n). (acesso direto a um elemento de uma matriz)

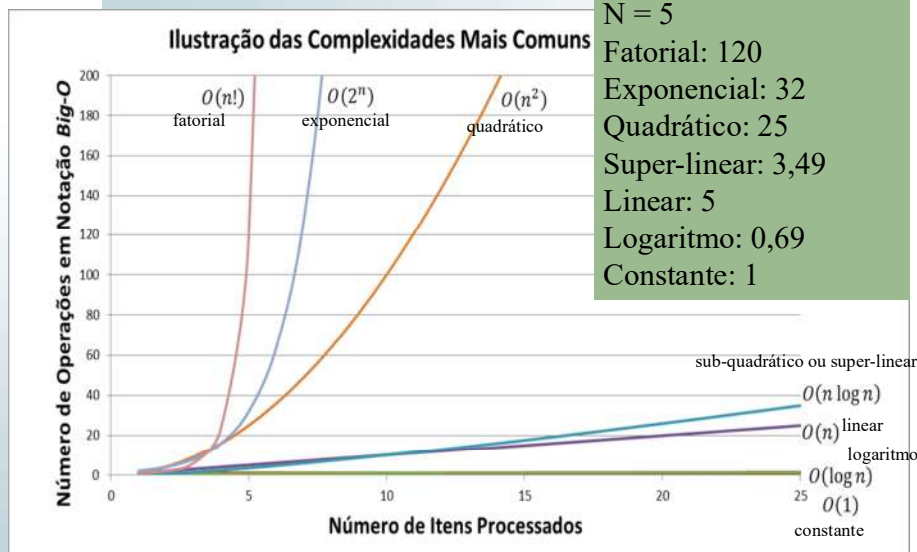


Métodos de Ordenação ou classificação

- Classificar um conjunto de dados consiste em arranjar seus elementos numa determinada ordem, segundo algum critério específico.
- Por exemplo, organizar em ordem alfabética um conjunto de dados alfanuméricos, ou então, ordenar de forma crescente ou decrescente dados numéricos.



Análise de Complexidade



Métodos de Ordenação ou classificação

- Método da Seleção
- Método da Inserção
- Método QuickSort
- Método BubbleSort (método da bolha)



Método da Seleção

- Ou ordenação por seleção, é um algoritmo de ordenação baseado em se passar sempre o menor valor do vetor para a primeira posição (ou o maior dependendo da ordem requerida), depois o de segundo menor valor para a segunda posição, e assim é feito sucessivamente com os (n-1) elementos restantes, até os últimos dois elementos.



Método da Inserção

- Simples algoritmo de ordenação, eficiente quando aplicado a um pequeno número de elementos. Em termos gerais, ele percorre um vetor de elementos da esquerda para a direita e à medida que avança vai deixando os elementos mais à esquerda ordenados.
- O algoritmo de inserção funciona da mesma maneira com que muitas pessoas ordenam cartas em um jogo de baralho como o pôquer.



Configuração inicial

G	J	E	H	F	C	D	A	I	B
---	---	---	---	---	---	---	---	---	---

animação

G J E H F C D A I B

Passo 1: coloca-se o menor elemento (A) na casa 1 (trocam-se as posições).

A J E H F C D G I B

Passo 2: coloca-se 2o. menor elemento (B) na casa 2.

A B E H F C D G I J

Passo 3: coloca-se o 3o. menor elemento (C) na casa 3.

A B C H F E D G I J

Passo 4: coloca-se o 4o. menor elemento (D) na casa 4.

A B C D F E H G I J

Passo 5: coloca-se o 5o. menor elemento (E) na casa 5.

A B C D E F H G I J

Passo 6: coloca-se o 6o. menor elemento (F) na casa 6.

A B C D E F H G I J

Passo 7: coloca-se o 7o. menor elemento (G) na casa 7.

A B C D E F G H I J

Passo 8: coloca-se o 8o. menor elemento (H) na casa 8.

A B C D E F G H I J

Passo 9: coloca-se o 9o. menor elemento (I) na casa 9.

A B C D E F G H I J



Método Quicksort

- Método de ordenação muito rápido e eficiente. O Quicksort adota a estratégia de divisão e conquista. Os passos são:
 - Escolha um elemento da lista, denominado pivô;
 - Rearranje a lista de forma que todos os elementos anteriores ao pivô sejam menores que ele, e todos os elementos posteriores ao pivô sejam maiores que ele. Ao fim do processo o pivô estará em sua posição final e haverá duas sublistas não ordenadas. Essa operação é denominada partição;
 - Recursivamente ordene a sublista dos elementos menores e a sublista dos elementos maiores;



Método Bubblesort (bolha)

- Algoritmo de ordenação dos mais simples.
- A ideia é percorrer o vetor diversas vezes, a cada passagem fazendo flutuar para o topo o menor elemento da sequência.



Lista de exercícios (08/12/22) dupla

- 1) Procurar mais dois métodos de ordenação e classificação
 - Citar, e explicar sobre os métodos
- 2) Simule o processo de classificação, mostrando cada passo, utilizando o princípio de funcionamento dos métodos de Seleção e Inserção para os vetores abaixo:

VI

05	09	02	06	11	07
----	----	----	----	----	----