Fatec
Araras

Estrutura de Dados

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br

“Ciência da Computação está tão relacionada aos computadores quanto a Astronomia aos telescópios, Biologia aos microscópios, ou Química aos tubos de ensaio.

A Ciência não estuda ferramentas. Ela estuda como nós as utilizamos, e o que descobrimos com elas.”

E. W. Dijkstra

Estrutura de Dados

Objetivo da aula

Fatec Araras

Conhecer Listas

- Tipos de Listas

Listas em C++

Listas em Python

BenchMark

Benchmark é o processo de avaliar o desempenho, eficiência ou confiabilidade de um sistema, componente ou processo, geralmente em comparação com um padrão conhecido ou com outros sistemas semelhantes.

Em computação, *benchmarking* é comumente usado para medir o desempenho de hardware, software, **algoritmos** ou sistemas em uma variedade de métricas, como velocidade de processamento, tempo de resposta, uso de memória, consumo de energia, entre outros.

BenchMark

Fatec Araras

Depois de entender como cada algoritmo de ordenação funciona, bserve o código **benckmark.cpp.**

Execute e anote o tempo de cada algoritmo apresentado:

#define MAX_SIZE 500

#define MAX_SIZE 5000

#define MAX_SIZE 50000

#define MAX_SIZE 500000

Qual é o algoritmo mais rápido ?

Notação Big O

Fatec Araras

A **notação Big O** é usada para analisar o desempenho de algoritmos e entender como o tempo de execução ou o espaço de memória necessário para executar um algoritmo aumenta à medida que o tamanho do problema (entrada) aumenta.

Em termos simples, a notação Big O descreve o pior caso de desempenho de um algoritmo. Ela fornece um limite superior assintótico para o tempo de execução (ou espaço de memória) em termos do tamanho da entrada.

Por exemplo, se um algoritmo tem uma complexidade de O(n), isso significa que o tempo de execução do algoritmo é linearmente proporcional ao tamanho da entrada. Se o tamanho da entrada dobrar, o tempo de execução também dobrará.

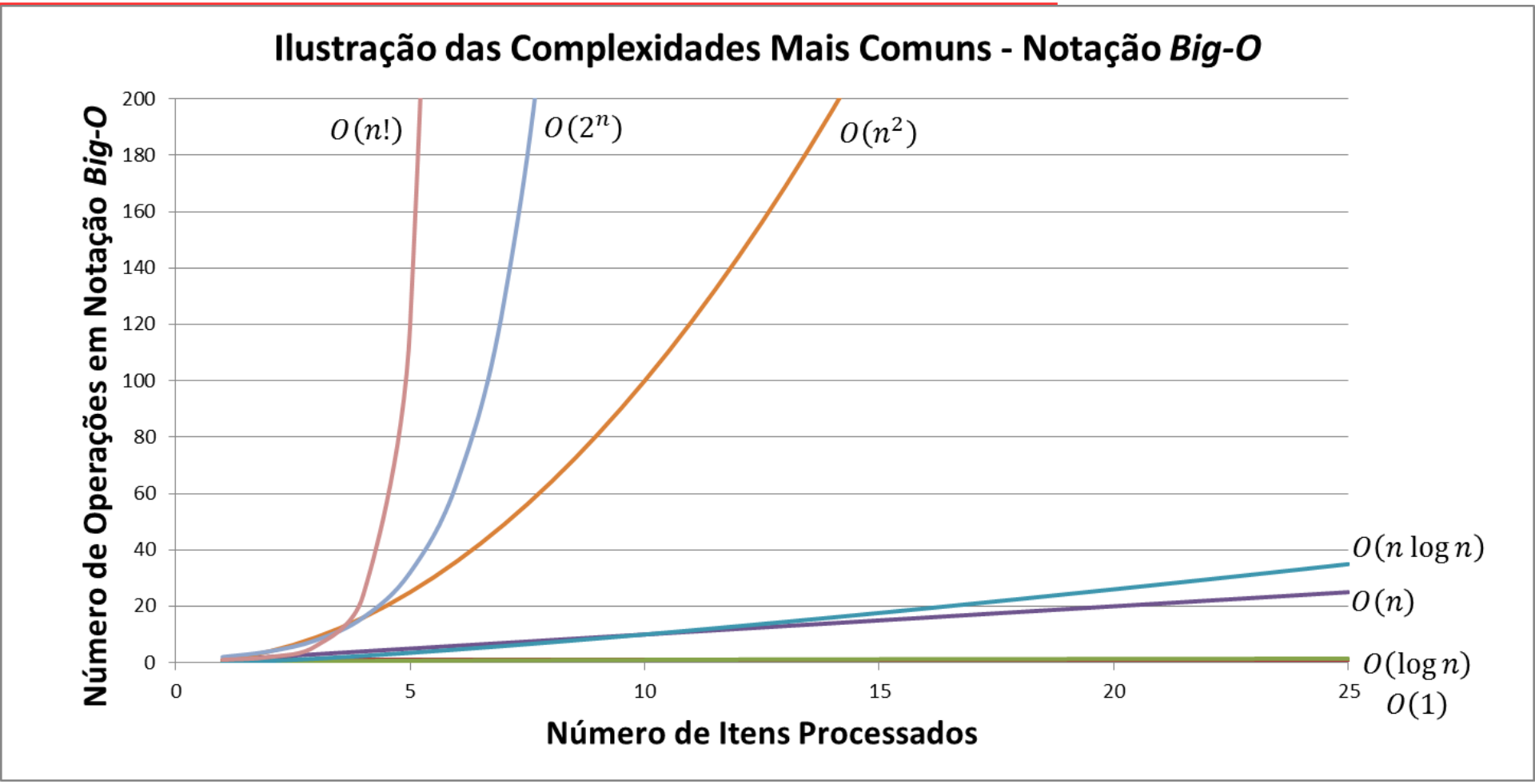
Notação Big O

Em termos simples, a notação Big O descreve o pior caso de desempenho de um algoritmo. Ela fornece um limite superior assintótico para o tempo de execução (ou espaço de memória) em termos do tamanho da entrada.

A notação Big O é frequentemente usada para classificar algoritmos com base em sua eficiência em relação ao tamanho da entrada.

Alguns exemplos comuns de notações Big O incluem O(1) para tempo de execução constante, O(log n) para tempo de execução logarítmico, O(n) para tempo de execução linear, O(n log n) para tempo de execução log-linear, O(n^2) para tempo de execução quadrático, entre outros.

Notação Big O



Fonte: https://pt.stackoverflow.com/questions/56836/defini%C3%A7%C3%A3o-da-nota%C3%A7%C3%A3o-big-o

BenchMark

Fatec Araras

A complexidade de tempo médio e pior caso para os algoritmos Quick Sort, Merge Sort e Bubble Sort são as seguintes:

Quick Sort:

Melhor caso: O(n log n) Caso médio: O(n log n) Pior caso: O(n^2)

Merge Sort:

Melhor caso: O(n log n) Caso médio: O(n log n) Pior caso: O(n log n)

Bubble Sort:

Melhor caso: O(n) Caso médio: O(n^2) Pior caso: O(n^2)

Arquivos

Fatec Araras

A maioria das linguagens de programação modernas possui mecanismos para manipular arquivos. No entanto, a forma como esses mecanismos são implementados pode variar de uma linguagem para outra.

•Em linguagens como C e C++, você pode usar as bibliotecas padrão <stdio.h> e <fstream> para manipular arquivos.

•Em Python, você pode usar as funções open() para abrir um arquivo e métodos como read(), write() e close() para manipular o conteúdo do arquivo.

•Em Java, você pode usar as classes File, FileReader, FileWriter, BufferedReader e BufferedWriter para manipular arquivos.

•Em JavaScript, você pode manipular arquivos usando APIs específicas do ambiente de execução, como o File API no navegador ou o módulo fs no Node.js.

Fatec
Araras

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br

Desafio

Fatec Araras

1) Crie um programa chamado writer.cpp. Este programa deve executar, lendo a entrada padrão até que o usuário digite 0.

Tudo que o usuário digitar, deve ser registrado no arquivo “desafio.txt”.

2) Crie um segundo programa chamado reader.cpp. Este deve ler todo o conteúdo do arquivo “desafio.txt” e apresentar na saída padrão.