

Universidade Federal de Viçosa
Campus Rio Paranaíba

“Nome” - “Matricula”

“TÍTULO DO PROJETO”

Rio Paranaíba - MG

2023

Universidade Federal de Viçosa
Campus **Rio Paranaíba**

“Nome” - “Matricula”

“TITULO DO PROJETO”

Trabalho apresentado para obtenção de créditos na disciplina SIN213 - Projeto de Algoritmo da Universidade Federal de Viçosa - Campus de Rio Paranaíba, ministrada pelo Professor Pedro Moisés de Souza.

Rio Paranaíba - MG

2023

1 RESUMO

Os algoritmos de ordenação apresentam grande importância no estudo teórico de algoritmos, bem como em aplicações rotineiras e práticas do cotidiano. Diante dessa grande importância ao longo dos anos surgem alguns algoritmos de ordenação na literatura, onde são analisados casos em que cada algoritmo se enquadra. Sendo assim se torna necessário um estudo para analisar o comportamento desses algoritmos, bem como suas complexidades para indicar qual é o mais adequado para cada tipo de problema.

...

Sumário

1	RESUMO	1
2	INTRODUÇÃO	3
3	ALGORITMOS	4
3.1	Insertion Sort	4
4	ANÁLISE DE COMPLEXIDADE	5
4.1	Insertion Sort	5
5	TABELA E GRÁFICO	6
5.1	Insertion Sort	6
6	CONCLUSÃO	8
7	REFERÊNCIAS BIBLIOGRÁFICAS	9
8	Função para calcular o tempo	10

2 INTRODUÇÃO

Em várias situações do nosso dia-a-dia nos deparamos com a necessidade de trabalharmos com dados/informações devidamente ordenadas, como, por exemplo, ao procurar um contato na lista telefônica, imagine como seria difícil se estes nomes não estivessem em ordem alfabética? Então não é difícil perceber que as atividades que envolvem algum método de ordenação estão muito presentes na computação.

...

3 ALGORITMOS

3.1 Insertion Sort

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Volutpat est velit egestas dui id ornare arcu. Sagittis purus sit amet volutpat consequat mauris nunc. Aliquet nec ullamcorper sit amet risus nullam eget felis. Fermentum odio eu feugiat pretium nibh. Velit euismod in pellentesque massa placerat dui ultricies. Ullamcorper morbi tincidunt ornare massa. Eget arcu dictum varius dui at consectetur lorem donec. Vitae ultricies leo integer malesuada nunc vel risus. Id aliquet risus feugiat in ante metus dictum. Nibh ipsum consequat nisl vel pretium. Auctor augue mauris augue neque gravida in fermentum. Id eu nisl nunc mi ipsum.

A lacus vestibulum sed arcu non odio euismod. Commodo ullamcorper a lacus vestibulum sed arcu non. Et pharetra pharetra massa massa ultricies mi quis. Ipsum a arcu cursus vitae congue mauris rhoncus aenean vel. Amet venenatis urna cursus eget nunc scelerisque. In cursus turpis massa tincidunt dui ut ornare lectus sit. Risus viverra adipiscing at in tellus. Nascetur ridiculus mus mauris vitae. In arcu cursus euismod quis viverra nibh cras. Duis ut diam quam nulla porttitor massa id neque aliquam. Volutpat blandit aliquam etiam erat velit scelerisque in. Erat velit scelerisque in dictum non consectetur a erat nam. Nisl nisi scelerisque eu ultrices vitae auctor eu augue ut. Facilisi etiam dignissim diam quis enim lobortis. Odio aenean sed adipiscing diam donec adipiscing.

Nullam non nisi est sit amet. Venenatis cras sed felis eget velit aliquet sagittis. Ut sem nulla pharetra diam sit amet nisl suscipit. Malesuada fames ac turpis egestas maecenas pharetra. Ultrices in iaculis nunc sed augue lacus viverra. Nisl condimentum id venenatis a. Venenatis lectus magna fringilla urna porttitor rhoncus dolor purus non. Amet volutpat consequat mauris nunc congue nisi vitae. Dolor sit amet consectetur adipiscing elit dui tristique sollicitudin nibh. Gravida quis blandit turpis cursus in hac habitasse. Enim sed faucibus turpis in eu mi. Dictum at tempor commodo ullamcorper a lacus vestibulum sed arcu.

4 ANÁLISE DE COMPLEXIDADE

4.1 Insertion Sort

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Volutpat est velit egestas dui id ornare arcu. Sagittis purus sit amet volutpat consequat mauris nunc. Aliquet nec ullamcorper sit amet risus nullam eget felis. Fermentum odio eu feugiat pretium nibh. Velit euismod in pellentesque massa placerat duis ultricies. Ullamcorper morbi tincidunt ornare massa. Eget arcu dictum varius duis at consectetur lorem donec. Vitae ultricies leo integer malesuada nunc vel risus. Id aliquet risus feugiat in ante metus dictum. Nibh ipsum consequat nisl vel pretium. Auctor augue mauris augue neque gravida in fermentum. Id eu nisl nunc mi ipsum.

A lacus vestibulum sed arcu non odio euismod. Commodo ullamcorper a lacus vestibulum sed arcu non. Et pharetra pharetra massa massa ultricies mi quis. Ipsum a arcu cursus vitae congue mauris rhoncus aenean vel. Amet venenatis urna cursus eget nunc scelerisque. In cursus turpis massa tincidunt dui ut ornare lectus sit. Risus viverra adipiscing at in tellus. Nascetur ridiculus mus mauris vitae. In arcu cursus euismod quis viverra nibh cras. Duis ut diam quam nulla porttitor massa id neque aliquam. Volutpat blandit aliquam etiam erat velit scelerisque in. Erat velit scelerisque in dictum non consectetur a erat nam. Nisl nisi scelerisque eu ultrices vitae auctor eu augue ut. Facilisi etiam dignissim diam quis enim lobortis. Odio aenean sed adipiscing diam donec adipiscing.

Nullam non nisi est sit amet. Venenatis cras sed felis eget velit aliquet sagittis. Ut sem nulla pharetra diam sit amet nisl suscipit. Malesuada fames ac turpis egestas maecenas pharetra. Ultrices in iaculis nunc sed augue lacus viverra. Nisl condimentum id venenatis a. Venenatis lectus magna fringilla urna porttitor rhoncus dolor purus non. Amet volutpat consequat mauris nunc congue nisi vitae. Dolor sit amet consectetur adipiscing elit duis tristique sollicitudin nibh. Gravida quis blandit turpis cursus in hac habitasse. Enim sed faucibus turpis in eu mi. Dictum at tempor commodo ullamcorper a lacus vestibulum sed arcu.

5 TABELA E GRÁFICO

5.1 Insertion Sort

	10	100	1000	10000	100000	1000000
Crescente	0	0	0	0	0	0
Decrescente	0	0	0	0	0	0
Aleatório	0	0	0	0		0

Figura 1: Tabela de tempo por segundo do algoritmo Insertion Sort

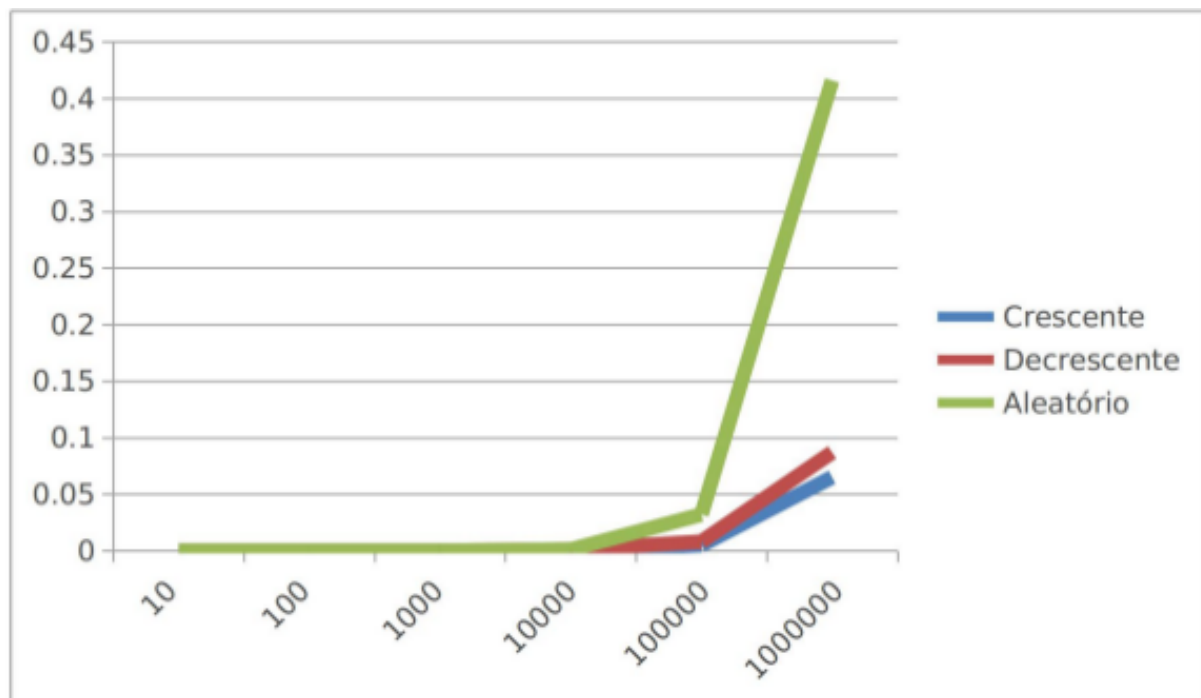


Figura 2: Gráfico de tempo por segundo do algoritmo Insertion Sort

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Mauris sit amet massa vitae tortor condimentum lacinia. Sit amet mauris commodo quis imperdiet massa tincidunt nunc pulvinar. Dictum fusce

ut placerat orci nulla pellentesque dignissim. Semper quis lectus nulla at volutpat diam ut. Urna condimentum mattis pellentesque id nibh. In nisl nisi scelerisque eu. Vestibulum mattis ullamcorper velit sed. Id velit ut tortor pretium. cursus eget nunc scelerisque viverra mauris. Lorem sed risus ultricies tristique nulla aliquet enim tortor at. Et leo duis ut diam quam. At erat pellentesque adipiscing commodo elit at imperdiet dui. Faucibus in ornare quam viverra. Sem integer vitae justo eget magna fermentum iaculis. Elit pellentesque habitant morbi tristique senectus. Elit duis tristique sollicitudin nibh. In nisl nisi scelerisque eu ultrices vitae. Nunc consequat interdum varius sit amet mattis vulputate enim nulla. Urna nunc id cursus metus aliquam.

6 CONCLUSÃO

Portanto, levando em consideração os estudos e os testes feitos nos três Métodos de Ordenação InsertionSort, SelectionSort e MergeSort ao longo deste trabalho pode-se perceber que em uma ordenação com 10 e 100 instâncias, os algoritmos possuem o mesmo desempenho, tanto para crescente, decrescente e randômico.

...

7 REFERÊNCIAS BIBLIOGRÁFICAS

8 Função para calcular o tempo

```
void operacoes(int tipo, int tamanho)
{
    clock_t start_t, end_t; //Variavel para guardar o tempo
    double tempoGasto;
    int * vetor = gerarSequencia(tipo, tamanho); //Gera sequencia de numeros
    //Salva a entrada de numeros
    salvarEntrada(tipo, tamanho, vetor);
    start_t = clock(); //Calcula o tempo atual
    insertionSort(vetor, tamanho); //Ordena o Vetor
    end_t = clock(); //Calcula o tempo apos ordenação
    tempoGasto = ((end_t - start_t) / (double)CLOCKS_PER_SEC); //Calcula diferença de
tempo
    salvarTempo(tipo, tamanho, tempoGasto); //Salva o tempo gasto
    //Salva a saída do programa
    salvarSaida(tipo, tamanho, vetor);
    //Libera a memoria
    free(vetor);
}
```