Instituto Politécnico de Beja

Estruturas de Dados e Algoritmos

Relatório

Docente: José Jasnau Caeiro

Abril de 2013

Índice

[Introdução 3](#_Toc353557402)

[Insertion-Sort 4](#_Toc353557403)

[Arquitetura do código 4](#_Toc353557404)

[Sistema de classes 4](#_Toc353557405)

[Variáveis escolhidas 4](#_Toc353557406)

[Métodos e Funções 4](#_Toc353557407)

[Módulos 4](#_Toc353557408)

[Desempenho computacional teoricamente previsto de cada algoritmo 4](#_Toc353557409)

[Projeto da experiência que permite obter os resultados para posterior análise experimental 4](#_Toc353557410)

[Gráﬁcos comparativos do desempenho computacional teoricamente esperado com os resultados experimentais 4](#_Toc353557411)

[Análise dos resultados experimentais obtidos 4](#_Toc353557412)

[Bubble-Sort 5](#_Toc353557413)

[Arquitetura do código 5](#_Toc353557414)

[Sistema de classes 5](#_Toc353557415)

[Variáveis escolhidas 5](#_Toc353557416)

[Métodos e Funções 5](#_Toc353557417)

[Módulos 5](#_Toc353557418)

[Desempenho computacional teoricamente previsto de cada algoritmo 5](#_Toc353557419)

[Projeto da experiência que permite obter os resultados para posterior análise experimental 5](#_Toc353557420)

[Gráﬁcos comparativos do desempenho computacional teoricamente esperado com os resultados experimentais 5](#_Toc353557421)

[Análise dos resultados experimentais obtidos 5](#_Toc353557422)

# Introdução

Este relatório aborda vários conceitos relacionados à complexidade de algoritmo, tendo como objetivo apresentar uma analise experimental englobando aspetos de diversos algoritmos de ordenação, tais como:

* Insertion-sort; ( Feito)
* Bubble-sort; ( Feito)
* Heapsort; ( Feito)
* Mergesort; ( Feito)
* Quicksort. ( Feito)

Este algoritmos de ordenação estão programados em linguagem java seguindo o pseudo-código subjacente presente no livro da disciplina.

Em relação ao algoritmos iremos mostrar ao longo do relatório o estudo dos mesmo, comparando o desempenho experimental computacional com as previsões teóricas.

Onde existe uma possível função de complexidade: **pior caso, melhor caso e caso médio**. Para classificar a ordem das funções temos três classificações: Ordem O, Ômega e Theta, que serão vistas ao longo do relatório.

Iremos estudar o código ao nível da sua arquitetura, sistema de classes, variáveis escolhidas, métodos e funções, módulos, desempenho computacional teoricamente previsto de cada algoritmo, os resultados para posterior analise experimental, comparação dos dados previsto com os dados obtidos e por fim a analise dos mesmos.

# Insertion-Sort

## Arquitetura do código

## Sistema de classes

## Variáveis escolhidas

## Métodos e Funções

## Módulos

## Desempenho computacional teoricamente previsto de cada algoritmo

## Projeto da experiência que permite obter os resultados para posterior análise experimental

## Gráﬁcos comparativos do desempenho computacional teoricamente esperado com os resultados experimentais

## Análise dos resultados experimentais obtidos

# Bubble-Sort

## Arquitetura do código

## Sistema de classes

## Variáveis escolhidas

## Métodos e Funções

## Módulos

## Desempenho computacional teoricamente previsto de cada algoritmo

## Projeto da experiência que permite obter os resultados para posterior análise experimental

## Gráﬁcos comparativos do desempenho computacional teoricamente esperado com os resultados experimentais

## Análise dos resultados experimentais obtidos