

# Algoritmos de Ordenação

## Papers and Abstracts

**Pedro Henrique Robadel da Silva Camâra<sup>1</sup>, Filipe Pinheiro da Costa Mello<sup>1</sup>**

<sup>1</sup>Departamento de Computação – Universidade Federal do Espírito Santo (UFES)  
R. Felício Alcure, Conceição – 29500-000 – Alegre – ES – Brasil

**Abstract.** *With an increasing volume of data, it's been made necessary to arrange them efficiently. With that being said, the huge data structure's sorting algorithms are essential for the field, evading the intuitive idea of rearranging in exchange of save time and computer performance, decreasing the computational cost. This article will cover bubble sort, insertion sort, binary insertion, shell sort, heap sort, quick sort\*, merge-sort, radix-sort and bucket sort analytically, using the data extracted from the code attached to the research and compare them by running time, number of comparisons and number of swaps. The goal is to conclude about speed and efficacy of each method for to observe in which environment each one will be better.*

*\*quick sort begin, center and median*

**Resumo.** *Com um volume grande de dados, se vê a necessidade de organizá-los de forma confiável e eficiente. Com isso, os algoritmos de ordenação de estruturas de dados grandes se fazem essenciais, saindo da ideia mais intuitiva de rearranjo para o maior ganho de tempo e processamento, diminuindo o custo computacional. Este trabalho irá abordar os algoritmos bolha, inserção direta, inserção binária, shellsort, heapsort, quicksort\*, merge-sort, radix-sort e bucketsort de forma crítica e analítica, utilizando dados gerados pelo compilador e pelo código anexado ao artigo para compará-los por tempo de execução, número de comparações e número de trocas. O objetivo deste estudo é chegar em uma conclusão sobre a velocidade e eficácia de cada algoritmo, a fim de observar em qual ambiente cada um se destaca.*

*\*quicksortini, quicksortcentro e quicksortmediana*

## Referências