



XVII ENCONTRO CIENTÍFICO DA UNDB
COMUNIDADES TRADICIONAIS: DESAFIOS E PERSPECTIVAS
(XVII EC 2024)

Algoritmos de Ordenação na Seleção Otimizada de Opções de Ação: aplicações no mercado financeiro¹

Bruno da Rocha Borges²

Luís Gabriel Fonseca Prado³

Mateus Verissimo Coelho⁴

Matheus Fernandes Barros⁵

Felipe Gomes Barbosa⁶

RESUMO

Neste estudo, têm-se como pretensão o uso de algoritmos de ordenação para otimizar a seleção de transações financeiras com base em critérios de atratividade definidos por equações específicas. Em tese, esses algoritmos podem reduzir o tempo de análise de forma significativa, o que melhora o processo de tomada de decisão no mercado financeiro. Dessa forma, para a comparação da eficiência de algoritmos distintos, foram realizados diferentes testes - tanto para algoritmos individuais quanto em híbridos - e avaliados com base no desempenho e eficiência de cada um em configurações de hardware variadas, em busca da escolha mais recomendada para ser utilizada no cotidiano do setor financeiro. Nesse contexto, a pesquisa apresenta conclusões retiradas a partir de testes e análises de adaptações com a combinação de algoritmos e códigos na linguagem de programação Python. Em conclusão, o seguinte estudo apresenta como a escolha do algoritmo adequado pode influenciar significativamente a agilidade e a eficácia na tomada de decisões financeiras.

Palavras-chave: Algoritmos de ordenação. Seleção de transações financeiras. Critérios de atratividade. Desempenho e eficiência.

1. INTRODUÇÃO

¹ Short Paper apresentado ao XVII Encontro Científico UNDB - Comunidades Tradicionais: Desafios e Perspectivas do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB.

² Graduando do 1º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: rock.borgs@gmail.com.

³ Graduando do 1º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: gabfonpr@gmail.com.

⁴ Graduando do 1º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: mateusversissimocoelho@gmail.com.

⁵ Graduando do 1º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: fer.matheusbarros@gmail.com.

⁶ Professor Felipe Gomes. Docente do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: felipe.barbosa@undb.edu.br.



XVII ENCONTRO CIENTÍFICO DA UNDB
COMUNIDADES TRADICIONAIS: DESAFIOS E PERSPECTIVAS
(XVII EC 2024)

A afirmação “Tempo é dinheiro”, é um dialeto popular comumente citado dentro do âmbito do mercado financeiro, ressaltando a importância da otimização do tempo em relação ao processo de tomada de decisões com o objetivo de gerar o máximo de lucro possível. Dentro do cenário tecnológico, uma forma de alcançar maior efetividade do uso do tempo no processo de tomada de decisões por meio da programação, é utilizando os algoritmos de ordenação, assim, possibilitando cumprir com o objetivo de otimizar a seleção de opções de ação, visando aprimorar a eficiência dentro do contexto do setor financeiro.

2. OBJETIVOS

Nessa análise, o seguinte projeto busca explorar a aplicação de algoritmos de ordenação para aumentar a eficiência no âmbito do mercado financeiro. Em um ambiente onde a velocidade e a precisão na ordenação de transações são cruciais nas tomadas de decisão, os algoritmos de ordenação se mostram ferramentas essenciais para a garantia de performance. Esse estudo visa o encontro e comparação de diferentes métodos de ordenação, como *Merge Sort*, *Quick Sort* e *Insertion Sort* em algoritmos híbridos, com o objetivo de identificar qual a melhor opção entre os modelos de ordenação para o contexto do mercado financeiro.

3. METODOLOGIA

Para identificar a eficiência de diferentes algoritmos de ordenação na classificação das transações financeiras, essa pesquisa utilizou uma abordagem de natureza básica, com a utilização de experimentos práticos, feitos com base em pesquisas literárias, com objetivo descritivo, observando o resultado de algoritmos distintos em diferentes hardwares, e abordagem quantitativa, que tudo pode ser mensurado.

Primeiramente, foram selecionados dois algoritmos de ordenação: *Merge Sort*, que utiliza a técnica ‘dividir e conquistar’, dividindo recursivamente uma lista em sublistas menores, ordenando-as, e em seguida as realocando em uma nova lista, mas dessa vez de forma ordenada. Por sua vez, o algoritmo *Quick sort*, também utiliza o método de “dividir e conquistar”, mas diferente do *Merge Sort*, ele seleciona um pivô como guia, para dividir a lista em sublistas menores se baseando no valor e posição do pivô escolhido (Nicácio; Sepulveda, 2019).

Posteriormente, com comparações realizadas entre os dois algoritmos de ordenação citados, surgiu a necessidade de otimizar ainda mais o desempenho no processo de ordenação das transações financeiras. Após análise do comportamento de ambos algoritmos, observou-se que, embora fossem altamente eficientes para grandes conjuntos de dados, existiam formas de otimizar sua performance a partir do momento que os algoritmos transformaram as grandes listas em sublistas menores.

De acordo com o que foi dito por Mohammad Rizal Hanafi (2022), o uso de algoritmos híbridos com o método *Insertion Sort* (que constroi uma lista ordenada gradualmente, inserindo um elemento de cada vez na posição correta), foram observadas como as melhores opções para a otimização desse processo, nesse caso, os algoritmos comparados foram: *Merge Sort* ou *Quick Sort* com *Insertion Sort*, essa combinação permite aproveitar a eficiência dos algoritmos principais nas partes maiores da lista, e nas menores, o *Insertion Sort*, que é mais rápido em listas pequenas, é utilizado para ordenar sublistas menores. Essa ação não apenas reduz o tempo total de execução, mas também melhora a eficiência do código como um todo.

Nesse contexto, a implementação dos algoritmos foi realizada em Python, para identificar rapidamente as transações financeiras mais atrativas. Foram utilizados algoritmos como *Merge Sort* e *Quick Sort* para dividir e ordenar grandes listas, avaliando sua eficiência para o contexto do estudo. Além disso, explorou-se o uso de algoritmos híbridos, aplicando *Insertion Sort* em sublistas menores para otimizar o tempo de execução. Listas de cem mil transações financeiras aleatórias foram criadas para simular valores reais, com variáveis como preço, volume, prazo de vencimento e spread. A atratividade de cada transação foi calculada usando a fórmula: $(opcao['volume'] / opcao['prazo_vencimento']) * (1 - opcao['spread']) * opcao['preco_exercicio']$ e posteriormente ordenada pelos métodos de ordenação aplicados.

Quadro 01 - Lista dos computadores utilizados para as comparações da pesquisa e suas especificações

Computador	Processador	Memória RAM	Sistema	Memória SSD
Computador 1	i7-8565U/1.80GHz	16GB	Windows 11	1T SSD
Computador 2	i5-11300H/3.10GHz	16GB	Windows 11	512 GB SSD
Computador 3	Ryzen 5 5500/3.60 GHz	16GB	Windows 11	240 GB SSD

Computador 4	Ryzen 5 55003.60 GHz	16GB	Windows 11	512 GB SSD
--------------	----------------------	------	------------	------------

4. RESULTADOS

Em função da finalidade da pesquisa atual, observou-se os códigos distintos e os seus resultados em todos os computadores, permitindo a análise e a comparação dos resultados obtidos, sendo eles os algoritmos individuais ou os algoritmos híbridos, e posteriormente definindo o melhor método de ordenação a ser utilizado no cenário financeiro.

Sob essa ótica, em relação aos algoritmos individuais, o *Quick Sort* apresentou a média de tempo de execução mais lenta, em listas com grande variação de valores, apresentado o seguinte gráfico a seguir:

Algoritmo (<i>Quick Sort</i>)	Tempo (em segundos)
Computador 1	0,4146
Computador 2	0,2954
Computador 3	0,2739
Computador 4	0,3522
Média total	0,3340

Posteriormente, ainda referente aos algoritmos individuais, o *Merge Sort* apresentou a segunda maior média de tempo de execução, em lista com grande variação de valores, apresentado o seguinte gráfico a seguir:

Algoritmo (<i>Merge Sort</i>)	Tempo (em segundos)
Computador 1	0,4543

Computador 2	0,3012
Computador 3	0,2639
Computador 4	0,3150
Média total	0,3336

Nessa análise, ao considerar os algoritmos híbridos, o *i* combinado com o *Insertion Sort*, apresentou uma melhora na média de tempo de execução quando comparado com os algoritmos de ordenação individuais. O algoritmo em questão foi o segundo melhor entre os analisados, apresentando o seguinte gráfico:

Algoritmo (<i>Quick Sort</i> + <i>Insertion Sort</i>)	Tempo (em segundos)
Computador 1	0,4110
Computador 2	0,2921
Computador 3	0,2613
Computador 4	0,3080
Média total	0,3181

Além disso, ainda em consideração dos algoritmos híbridos, o *Merge Sort* combinado com o *Insertion Sort* apresentou a melhor média de tempo de execução entre os algoritmos analisados. O algoritmo em questão foi o segundo melhor entre os analisados, apresentando o seguinte gráfico:

Algoritmo (<i>Merge Sort</i> + <i>Insertion Sort</i>)	Tempo (em segundos)
--	----------------------------

Computador 1	0,2830
Computador 2	0,1959
Computador 3	0,1684
Computador 4	0,3038
Média total	0,2378

Nesse contexto, foram analisados os testes e as revisões dos gráficos de cada algoritmo, permitindo identificar o método mais eficiente de ordenação para decisões financeiras em situações reais, onde é apenas necessário adaptar o código para usar dados reais de uma empresa, em vez de opções aleatórias, calculando e ordenando os elementos da lista com base na atratividade de cada um.

5. CONCLUSÃO

Portanto, os testes realizados neste estudo demonstraram que, no contexto do mercado financeiro, onde a velocidade de processamento é fundamental, o desempenho dos algoritmos de ordenação se torna um fator determinante na escolha do método a ser utilizado. Os testes realizados permitiram a demonstração da eficiência dos algoritmos híbridos em detrimento dos individuais, onde houve o aproveitamento do *Merge Sort* ou do *Quick Sort* em listas maiores, enquanto o *Insertion Sort* foi eficaz na ordenação de sublistas menores, resultando em um desempenho superior entre os algoritmos avaliados.

Por fim, esta pesquisa evidencia a importância de considerar tanto as características dos dados quanto o ambiente de execução ao selecionar algoritmos de ordenação. Recomenda-se a continuidade das pesquisas nesta área, explorando novas combinações de métodos de ordenação para serem executados em diferentes cenários de mercado. Tais avanços poderão fortalecer ainda mais o processo decisório no setor, tornando-o mais ágil e preciso.

Em resumo, a escolha adequada e a combinação de forma eficaz de algoritmos de ordenação são essenciais para otimizar o processo decisório no mercado financeiro, fornecendo vantagens competitivas significativas.



XVII ENCONTRO CIENTÍFICO DA UNDB
COMUNIDADES TRADICIONAIS: DESAFIOS E PERSPECTIVAS
(XVII EC 2024)

REFERÊNCIAS

NICÁCIO, J.; SEPULVEDA, G. **Anais da VI Semana Tecnológica Acadêmica de Ciência da Computação da UTFPR, câmpus Santa Helena**. Santa Helena: UTFPR, 2019. Disponível em: <http://sh.utfpr.edu.br/setac/anais2019.pdf#page=43>. Acesso em: 1 nov. 2024.

HANAFI, M. R. et al. **Comparison Analysis of Bubble Sort Algorithm with Tim Sort Algorithm Sorting Against the Amount of Data**. Journal of Computer Engineering, Electronics and Information Technology, v. 1, n. 1, p. 29–38, 2022. Disponível em: <https://ejournal.upi.edu/index.php/COELITE/article/view/43794/18701>. Acesso em: 1 nov. 2024.



XVII ENCONTRO CIENTÍFICO DA UNDB
COMUNIDADES TRADICIONAIS: DESAFIOS E PERSPECTIVAS
(XVII EC 2024)