Relatório Técnico

Algoritmo de Ordenação

Amanda N. Kuehn1

1Graduanda em Sistemas de Informação - UNEX

amanda.kuehn@ftc.edu.br

**Resumo.** O presente relatório detalha e justifica a escolha do algoritmo de ordenação Counting Sort para a empresa Manga Rosa. Além de descrever a solução implementada, realizar comparações com outros métodos de ordenação e apresentra considerações finais sobre a eficácia do Counting Sort no contexto da empresa.

# 1. Introdução

A Manga Rosa, uma empresa que processa milhares de pedidos diariamente em setores diversos, enfrenta desafios significativos em sua infraestrutura logística devido ao alto volume de dados que precisam ser processados e ordenados eficientemente. Atualmente, a empresa utiliza um algoritmo de ordenação com complexidade temporal de O(n log n) melhores, piores e casos médios. Esse tipo de complexidade é comum em algoritmos clássicos de ordenação, como QuickSort, MergeSort e HeapSort.

O QuickSort, por exemplo, é amplamente utilizado devido à sua eficiência média e à simplicidade de implementação. Ele possui complexidade de tempo O(n log n) na maioria dos casos, mas pode ter desempenho O(n2) no pior caso, especialmente quando os dados estão quase ordenados ou todos os elementos são iguais. O MergeSort também tem complexidade O(n log n) em todos os casos e é mais estável, mas requer espaço adicional proporcional ao número de elementos, o que pode ser uma limitação para grandes volumes de dados. Já o HeapSort é um algoritmo de ordenação por seleção que possui uma complexidade de O(n log n) e não necessita de espaço adicional, mas tende a ser mais lento na prática devido às operações de heapify.

Embora eficazes, esses algoritmos apresentam limitações ao lidar com o grande volume de dados gerados pela Manga Rosa, especialmente considerando a necessidade de operações rápidas e eficientes em um ambiente de alta demanda. Este cenário é particularmente desafiador para clientes com grandes volumes de pedidos, como os marketplaces, que dependem de relatórios ordenados para otimizar suas operações logísticas.

Diante desse desafio, o algoritmo de ordenação escolhido para resolver o problema foi o Counting Sort. Este algoritmo, conhecido por sua complexidade temporal linear em termos de tempo, mostrou-se uma solução ideal para o tipo de dados e a escala de operações realizadas pela empresa. O Counting Sort não apenas atende aos requisitos de complexidade temporal e espacial, mas também é particularmente eficiente para ordenar strings alfanuméricas de comprimento fixo, como as utilizadas pela Manga Rosa.

O presente relatório será dividido em seis seções: a seção 1 traz uma introdução ao tema; a seção 2 discorre sobre a justificativa da escolha do algoritmo; a seção 3 descreve a solução utilizada frente ao problema proposto; a seção 4 apresenta a comparação entre algoritmos; a seção 5 expõe um gráfico das comparações apresentadas na seção anterior; e a seção 6 refere-se às considerações finais.

# 2. Justificativa

Como afirmado anteriormente, o algoritmo de ordenação escolhido foi o Counting Sort. O principal critério para a escolha do algoritmo foi a necessidade de uma complexidade temporal linear em todos os casos (melhor, pior e médio). O Counting Sort se destaca por sua capacidade de ordenar elementos em tempo O(n + k), onde n é o número de elementos e k é o intervalo de valores possíveis. No caso específico da Manga Rosa, onde os códigos alfanuméricos têm um tamanho fixo de 13 caracteres, k pode ser tratado como uma constante, o que simplifica a complexidade temporal para O(n). Essa eficiência é crucial para lidar com o grande volume de dados processados diariamente pela empresa, garantindo que a ordenação não se torne um gargalo no fluxo operacional.

# 3. Solução proposta

A solução proposta para o desafio da Manga Rosa envolve a implementação do algoritmo Counting Sort para ordenar eficientemente os pedidos identificados por códigos alfanuméricos de 13 caracteres. O Counting Sort é uma escolha estratégica devido à sua capacidade de lidar com grandes volumes de dados de forma rápida e eficiente.

O Counting Sort possui uma complexidade de tempo de O(n \* k), onde n é o número de elementos (ou seja, a quantidade de pedidos) e k é o comprimento das strings (neste caso, 13 caracteres). Como o comprimento das strings é fixo e pequeno, k pode ser tratado como uma constante, simplificando a complexidade para O(n). Isso significa que o tempo necessário para ordenar os pedidos cresce linearmente com o número de pedidos, independentemente de como esses pedidos estão inicialmente organizados. Essa característica é especialmente vantajosa para a Manga Rosa, que processa um grande volume de pedidos diariamente.

A complexidade espacial do Counting Sort é O(n + k), o que significa que ele requer espaço adicional proporcional ao número de pedidos e ao comprimento das strings. No contexto da Manga Rosa, onde os códigos alfanuméricos têm um comprimento fixo, esse espaço adicional é gerenciável e adequado. Comparado a outros algoritmos de ordenação com complexidade espacial superior, o Counting Sort oferece uma solução eficiente em termos de uso de memória, mantendo o consumo dentro dos limites aceitáveis para grandes volumes de dados.

A implementação do Counting Sort foi especificamente adaptada para lidar com strings alfanuméricas de comprimento fixo. Ao invés de realizar comparações entre os elementos, o Counting Sort conta a frequência de cada caractere em cada posição da string e, em seguida, utiliza essas contagens para determinar a posição final de cada elemento na lista ordenada. Essa abordagem elimina a necessidade de múltiplas comparações, resultando em um processo de ordenação mais rápido e eficiente, especialmente quando comparado a algoritmos com complexidade O(n log n).

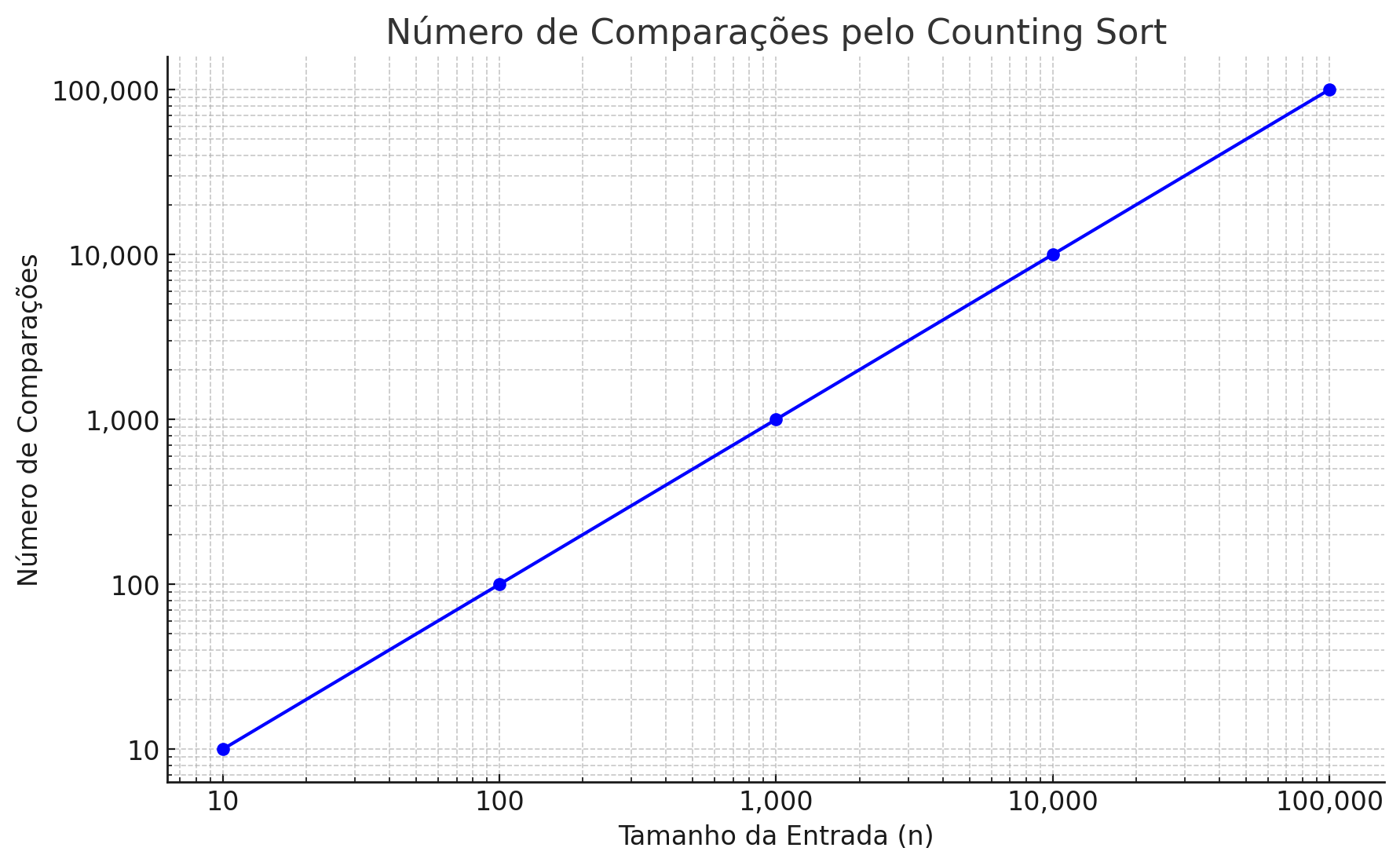
A solução baseada no Counting Sort oferece uma série de vantagens para a Manga Rosa. Primeiramente, a ordenação é realizada em tempo linear, permitindo que a empresa processe grandes volumes de pedidos rapidamente. Além disso, o consumo de memória é controlado e proporcional ao número de pedidos, tornando-o adequado para o tipo de dados tratado. Finalmente, o algoritmo é especialmente eficaz para ordenar strings alfanuméricas de tamanho fixo, como os códigos de 13 caracteres utilizados pela empresa.

# 4. Comparação

O algoritmo anterior tem como pontos positivos a eficiência média de O(n log n) e sua ampla aceitação e compreensão. No entanto, apresenta desvantagens, como menor eficiência para dados com comprimento fixo e maior consumo de memória e tempo.

O algoritmo atual, Counting Sort, oferece a vantagem de uma complexidade linear O(n), adequada para dados com intervalo fixo e pequeno, e reduzido consumo de memória adicional. Contudo, é menos flexível para dados com comprimento variável ou intervalo muito grande.

# 5. Gráfico de Comparações



**Gráfico 1. Comparações.**

O gráfico acima mostra o número de comparações realizadas pelo algoritmo Counting Sort para diferentes tamanhos de entrada. Como esperado, o número de comparações cresce linearmente com o aumento do tamanho da entrada, refletindo a complexidade de tempo O(n) do Counting Sort. Esse comportamento confirma a eficiência do algoritmo para grandes volumes de dados, como os processados pela Manga Rosa.

# 6. Considerações Finais

Em resumo, o Counting Sort foi selecionado pela sua eficiência e adequação ao cenário específico da Manga Rosa, onde grandes volumes de dados precisam ser ordenados de forma rápida e eficaz. A análise revelou que, para dados com intervalos e tamanhos fixos, como os códigos alfanuméricos da empresa, o Counting Sort oferece vantagens significativas sobre algoritmos de ordenação mais tradicionais, que possuem complexidade de tempo O (n log n). Com a implementação desse algoritmo, espera-se que a Manga Rosa otimize seus processos de ordenação, melhorando assim a sua capacidade de resposta e a eficiência operacional.

# References

Bajpai, K. and Kots, A. (2014) “Implementing and Analyzing an Efficient Version of Counting Sort (E-Counting Sort)”, *International Journal of Computer Applications*, Vol. 98, No. 9, pp. 1-2.

Hoare, C. A. R. (1961) “Quicksort”, *In: Communications of the ACM, Proceedings of the ACM National Meeting*, New York: ACM Press, pp. 321-322.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2012) *Algoritmos: teoria e prática*. 3rd ed., Rio de Janeiro: Elsevier.