

UFSM
CIÊNCIA DA COMPUTAÇÃO
PESQUISA E ORDENAÇÃO DE DADOS

LEANDRO OLIVEIRA DO NASCIMENTO

Algoritmos de ordenação: Merge Insertion Sort e Bucket Sort com Merge Insertion Sort

Algoritmos de ordenação: Merge Insertion Sort e Bucket Sort com Merge Insertion Sort

A escolha de implementar algoritmos de ordenação híbridos muitas vezes surge da necessidade de combinar as vantagens de diferentes abordagens para otimizar o desempenho e a eficiência em diferentes cenários de aplicação. Nesse contexto, foram implementados o Merge Insertion Sort e o Bucket Sort com Merge Insertion Sort, cada um com características específicas que os tornam adequados para diferentes situações.

Merge Insertion Sort:

O Merge Insertion Sort é uma combinação entre o Merge Sort e o Insertion Sort. O Merge Sort é conhecido por sua eficiência em lidar com grandes conjuntos de dados, dividindo-os recursivamente em subconjuntos menores, ordenando-os e, em seguida, mesclando esses subconjuntos ordenados. Por outro lado, o Insertion Sort é eficiente em lidar com conjuntos de dados pequenos ou quase ordenados, pois realiza inserções diretas nos elementos do array. Ao combinar essas duas abordagens, o Merge Insertion Sort pode tirar proveito da eficiência do Merge Sort em conjuntos maiores e da eficiência do Insertion Sort em conjuntos menores.

Bucket Sort com Merge Insertion Sort:

O Bucket Sort é um algoritmo de ordenação que divide o conjunto de dados em "baldes" com base em intervalos de valores e, em seguida, ordena cada balde separadamente. Uma vez que os baldes estão ordenados, os elementos são combinados para formar o conjunto de dados ordenado. O Bucket Sort é especialmente eficiente quando os dados estão uniformemente distribuídos entre os baldes. Ao combinar o Bucket Sort com o Insertion Sort para ordenar os baldes individualmente, é possível aproveitar a eficiência do Insertion Sort em conjuntos menores.

Quando um balde contém uma grande quantidade de dados, o uso do Insertion Sort pode se tornar menos eficiente devido à sua complexidade de tempo quadrático. Nesses casos, é aplicado o algoritmo de mesclagem (merge) para ordenar os elementos do balde de forma mais eficiente. Essa estratégia permite lidar de forma eficaz com conjuntos de dados heterogêneos, onde alguns baldes podem conter uma quantidade significativa de elementos, enquanto outros podem conter muito menos.

Escolha da Implementação Híbrida:

A escolha de implementar algoritmos híbridos como o Merge Insertion Sort e o Bucket Sort com Merge Insertion Sort pode ser motivada pela necessidade de lidar com conjuntos de dados de diferentes tamanhos e distribuições. Enquanto o Merge Insertion Sort pode ser mais adequado para conjuntos de dados maiores, devido à sua capacidade de divisão e mesclagem eficientes, o Bucket Sort com Insertion Sort

pode se destacar em conjuntos de dados distribuídos de forma mais uniforme entre os baldes.

Eficiência dos Algoritmos

Para conjuntos de dados volumosos, acima de 200 mil elementos, na ordenação externa, o Bucket com Merge Insertion Sort(miSort) demonstrou superioridade em relação ao Merge Insertion Sort puro. Ele se revelou ligeiramente mais rápido na maioria dos casos, exibindo uma estabilidade consistente no tempo de execução. Embora o Merge com Insertion Sort tenha se destacado em alguns cenários, foi geralmente superado pelo Bucket com Merge Insertion Sort.

Por outro lado, em conjuntos menores, abaixo de 200 mil elementos, tanto na ordenação interna quanto externa, o Merge com Insertion Sort mostrou-se mais eficiente na maioria das comparações. Entretanto, é importante notar que o Bucket com Insertion Sort frequentemente apresentou um desempenho superior na maioria das ordenações externas, enquanto o Merge Insertion Sort se destacou em grande parte das situações de ordenação interna.

Além disso, é importante destacar que nos testes realizados com mais de 200 mil elementos, foram feitos apenas para a ordenação externa. Talvez por essa razão, o Bucket Merge Insertion Sort tenha demonstrado um desempenho superior, uma vez que o Merge Insertion Sort parece ser mais adequado para a ordenação interna, enquanto o Bucket Merge Insertion Sort se destaca mais na ordenação externa.

Ordenação Externa:

Para a ordenação externa, além dos algoritmos híbridos, foi empregado o Merge Sort Externo. Como os algoritmos híbridos implementados são destinados à ordenação interna, o **Merge Sort Externo** foi utilizado para dividir os dados em arquivos, seguido pela aplicação dos algoritmos híbridos para ordenar internamente esses arquivos. Posteriormente, os arquivos ordenados foram mesclados em um único arquivo, resultando em uma ordenação completa.

Teste do Tempo de Execução

Tamanho da Ordenação: 100

1 - MergeSort com InsertionSort X

Tempo de execução do algoritmo de ordenação externo: 0.000000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.000000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.008000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.000000 segundos

2 - BucketSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 0.000000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.130000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.000000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.118000 segundos

Tamanho da Ordenação: 1000

1 - MergeSort com InsertionSort X

Tempo de execução do algoritmo de ordenação externo: 0.005000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.000000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.012000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.000000 segundos

2 - BucketSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 0.009000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.136000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.004000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.123000 segundos

Tamanho da Ordenação: :10 000

1 - MergeSort com InsertionSort X

Tempo de execução do algoritmo de ordenação externo: 0.053000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.001000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.037000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.001000 segundos

2 - BucketSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 0.033000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.240000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.038000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.165000 segundos

Tamanho da Ordenação: 50 000

1 - MergeSort com InsertionSort X

Tempo de execução do algoritmo de ordenação externo: 0.148000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.005000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.148000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.010000 segundos

2 - BucketSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 0.133000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.216000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.149000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.118000 segundos

Tamanho da Ordenação: 100 000

1 - MergeSort com InsertionSort X

Tempo de execução do algoritmo de ordenação externo: 0.318000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.007000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.291000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.008000 segundos

2 - BucketSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 0.271000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.114000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.291000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.197000 segundos

Tamanho da Ordenação: 200 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 0.662000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.017000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.646000 segundos

Tempo de execução do algoritmo de ordenação interno: 0.017000 segundos

2 - BucketSort com InsertionSort X

Tempo de execução do algoritmo de ordenação externo: 0.564000 segundos
Tempo de execução do algoritmo de ordenação interno: 0.008000 segundos

Tempo de execução do algoritmo de ordenação externo: 0.608000 segundos
Tempo de execução do algoritmo de ordenação interno: 0.010000 segundos

Tamanho da Ordenação: 350 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 1.031000 segundos
Tempo de execução do algoritmo de ordenação interno: 1.161000 segundos

2 - BucketSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 0.969000 segundos
Tempo de execução do algoritmo de ordenação interno: 1.105000 segundos

Tamanho da Ordenação: 500 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 1.382000 segundos
Tempo de execução do algoritmo de ordenação interno: 1.251000 segundos

2 - BucketSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 1.252000 segundos
Tempo de execução do algoritmo de ordenação interno: 1.225000 segundos

Tamanho da Ordenação: 700 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 1.962000 segundos
Tempo de execução do algoritmo de ordenação interno: 1.775000 segundos

2 - BucketSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 1.825000 segundos
Tempo de execução do algoritmo de ordenação interno: 1.895000 segundos

Tamanho da Ordenação: 950 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 2.517000 segundos
Tempo de execução do algoritmo de ordenação interno: 2.484000 segundos

2 - BucketSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 2.396000 segundos
Tempo de execução do algoritmo de ordenação interno: 2.487000 segundos

Tamanho da Ordenação: 1 500 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 3.880000 segundos

Tempo de execução do algoritmo de ordenação externo: 4.372000 segundos

2 - BucketSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 3.882000 segundos

Tempo de execução do algoritmo de ordenação externo: 3.827000 segundos

Tamanho da Ordenação: 4 000 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 11.028000 segundos

Tempo de execução do algoritmo de ordenação externo: 12.031000 segundos

2 - BucketSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 11.341000 segundos

Tempo de execução do algoritmo de ordenação externo: 11.225000 segundos

Tamanho da Ordenação: 7 000 000

1 - MergeSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 20.240000 segundos

Tempo de execução do algoritmo de ordenação externo: 21.304000 segundos

2 - BucketSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 21.313000 segundos

Tempo de execução do algoritmo de ordenação externo: 20.499000 segundos

Tamanho da Ordenação: 10 000 000

1 - MergeSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 30.066000 segundos

Tempo de execução do algoritmo de ordenação externo: 29.919000 segundos

2 - BucketSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 30.833000 segundos

Tempo de execução do algoritmo de ordenação externo: 30.191000 segundos

Tamanho da Ordenação: 15 000 000

1 - MergeSort com InsertionSort

Tempo de execução do algoritmo de ordenação externo: 43.201000 segundos

Tempo de execução do algoritmo de ordenação externo: 42.565000 segundos

2 - BucketSort com InsertionSort **X**

Tempo de execução do algoritmo de ordenação externo: 42.697000 segundos

Tempo de execução do algoritmo de ordenação externo: 42.837000 segundos