



UNIVERSIDADE FEDERAL RURAL DO SEMI - ÁRIDO - UFRSA
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
ESTRUTURA DE DADOS I

COMB SORT

Algoritmo de ordenação

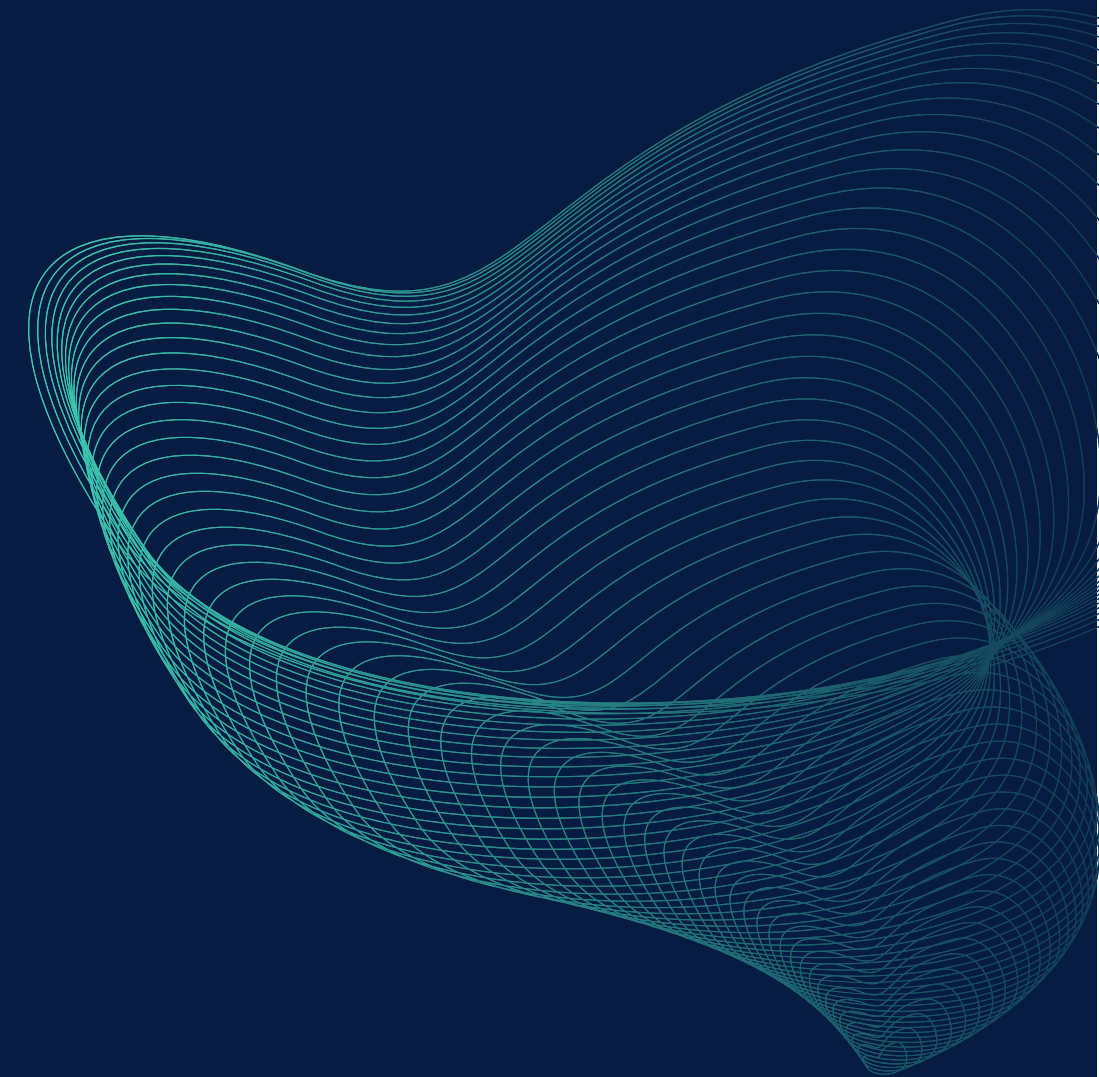
Integrantes:

Ana Vitória de Queiroz Oliveira

Laura Gonçalves Fernandes

O que é Comb Sort?

- O Comb Sort é um algoritmo de ordenação que compara e troca elementos em um array, com o objetivo de organizá-los em ordem.
- Utiliza um conceito de "gaps" entre os elementos, que são reduzidos gradualmente até que o array esteja totalmente ordenado



Funcionamento do Comb Sort

Início

O array é dividido em segmentos com uma determinada distância (gap) entre eles.

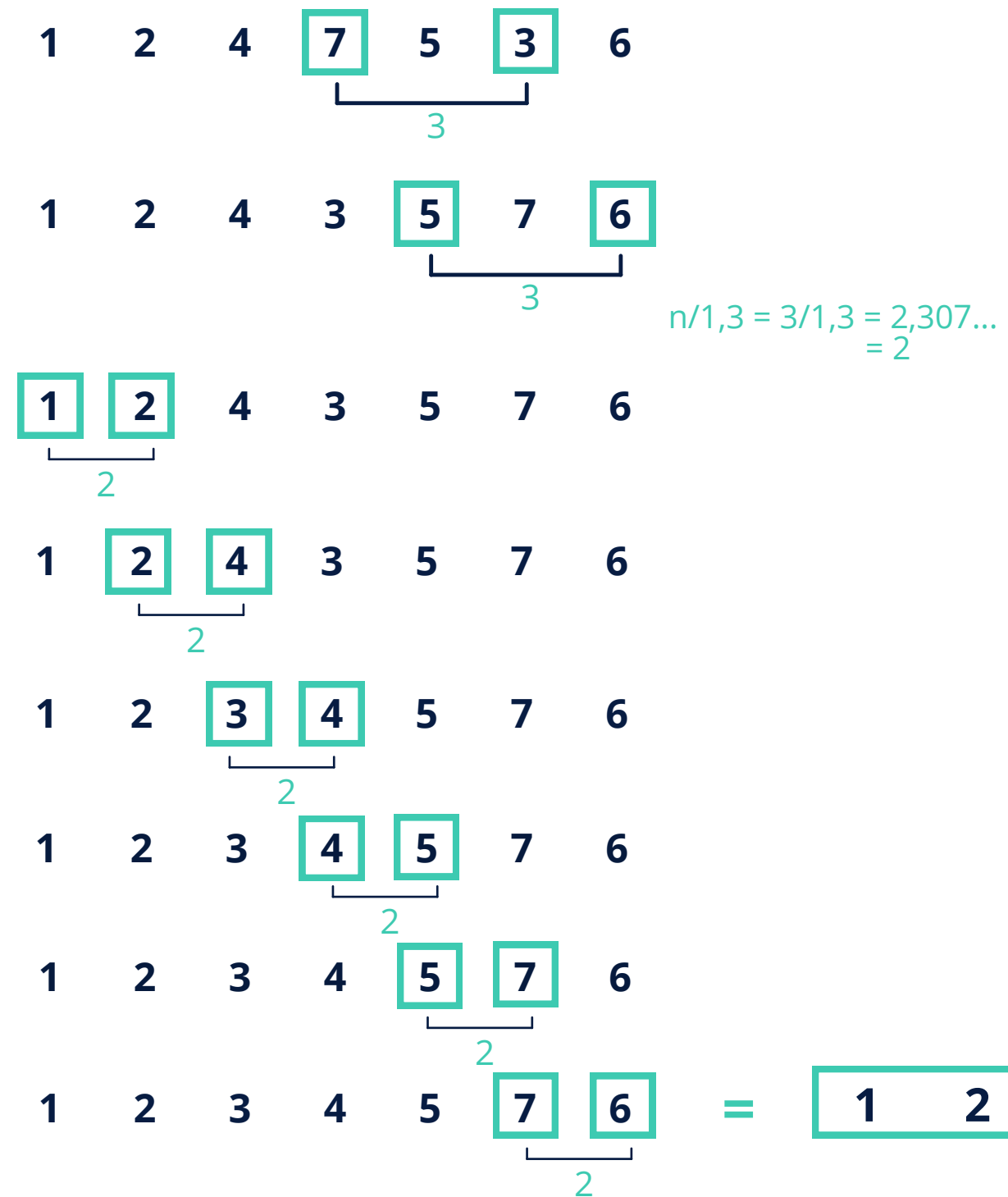
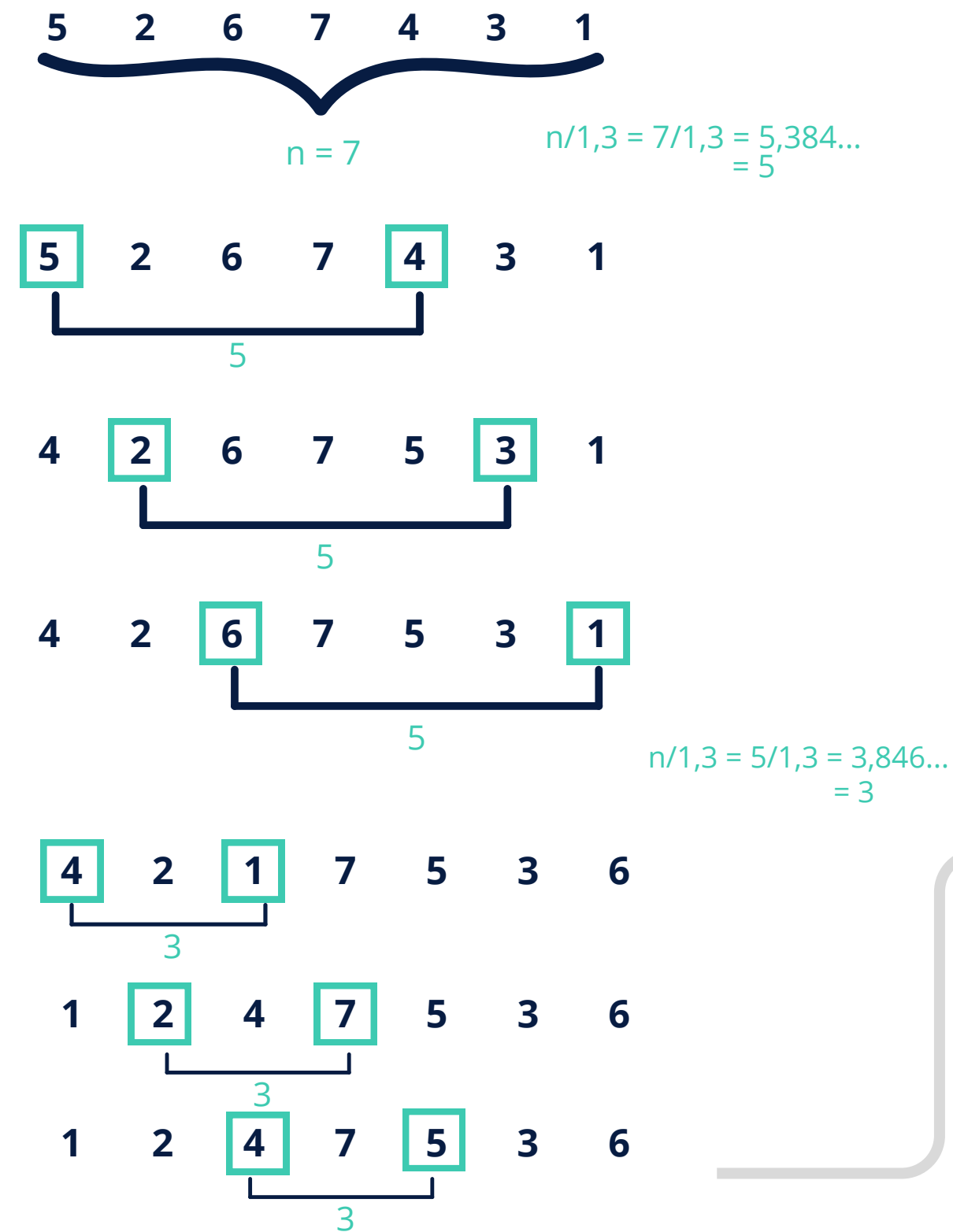
Gap inicial: tamanho da lista / 1,3

Comparação

Os elementos separados pelo gap são comparados e, se necessário, trocados de posição.

Redução do gap

A cada iteração, o gap é reduzido, geralmente dividindo-o por um fator (como 1.3), até que eventualmente o gap seja 1, momento em que o algoritmo funciona como um bubble sort.



Exemplo de Comb Sort

Vantagens do Comb Sort



Melhor desempenho

O Comb Sort tem um desempenho significativamente melhor do que o Bubble Sort, especialmente em arrays grandes.

Adaptabilidade

O algoritmo se adapta automaticamente ao tamanho e complexidade do array, ajustando o gap de forma eficiente.

Simplicidade

A implementação do Comb Sort é relativamente simples, tornando-o fácil de entender e implementar.

Escalabilidade

O Comb Sort pode ser aplicado a arrays de qualquer tamanho, sem comprometer sua eficiência.



Ineficiente para Listas Quase Ordenadas

Pode ser menos eficiente em listas que já estão quase ordenadas. O algoritmo pode fazer comparações desnecessárias, resultando em um desempenho inferior ao de algoritmos otimizados para listas quase ordenadas, como o Insertion Sort.

Desempenho no Pior Caso

A complexidade de tempo no pior caso do Comb Sort é $O(n^2)$, semelhante ao Bubble Sort. Isso significa que, em situações desfavoráveis, ele pode ser significativamente mais lento do que algoritmos como Quick Sort ou Merge Sort, que têm uma complexidade média de $O(n \log n)$.

Desvantagens do Comb Sort

Dependência do Fator de Redução

O desempenho do Comb Sort pode ser sensível à escolha do fator de redução (geralmente 1.3). Se o fator não for bem escolhido, pode não alcançar o desempenho esperado.

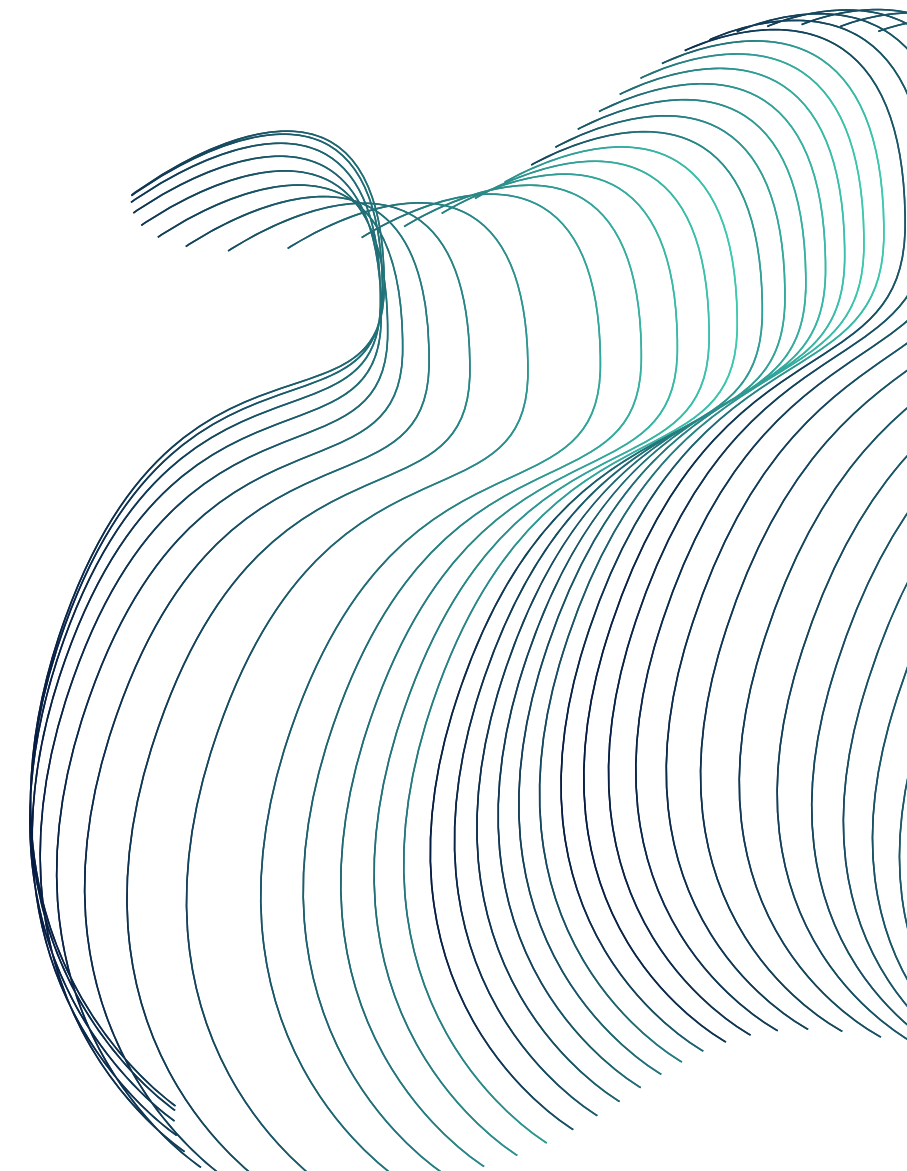
Comparação com outros algoritmos ordenação

Quick Sort e Merge Sort

Embora o Comb Sort seja mais eficiente que o Bubble Sort, ele ainda é superado por algoritmos de ordenação mais avançados, como o Quicksort e o Mergesort

Bubble Sort

O Comb Sort é uma melhoria significativa do Bubble Sort, com um desempenho muito superior.



Aplicabilidade do Comb Sort

Processamento de Dados

Útil em sistemas que lidam com grandes volumes de dados, como bancos de dados e sistemas de armazenamento.



Análise Financeira

Ele pode ser aplicado em cenários financeiros, como análise de portfólio e processamento de transações.

Computação em Nuvem

Em sistemas de computação distribuída, o Comb Sort pode ser usado para ordenar e organizar dados em larga escala.



Manipulação de Imagens

Encontra aplicações em processamento de imagens, como ordenação de pixels e organização de metadados.

Complexidade do Comb Sort

Espaço:

Realiza a ordenação diretamente no array original sem necessitar de memória extra proporcional ao tamanho da entrada. Isso torna o Comb Sort eficiente em termos de uso de espaço, utilizando apenas **$O(1)$** de espaço auxiliar.

Tempo:

Melhor caso:

$$O(n \log n)$$

Médio caso

$$O(n^2/2)$$

Pior caso

$$O(n^2)$$

