



Projeto e Análise de Algoritmos

TRABALHO 1

PROPOSTA 1 – Leonardo Vanzin, Lucas Veit de Sá e Roberta Aparecida da Silva Alcantara

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método quadrático BubbleSort e analisar a possibilidade de otimização do funcionamento básico do algoritmo.

Tarefa:

Implementar as seguintes estratégias de ordenação do BubbleSort:

- Abordagem clássica em que o método tem todas as iterações executadas normalmente, sem nenhum tipo de otimização;
- Abordagem com uma flag que identifica se foi realizada alguma operação de swap e, em caso negativo, termina a execução do método sem a necessidade de executar as iterações restantes;

Deve-se avaliar o comportamento dos métodos perante conjuntos de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TnsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- Os dois métodos de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todos os métodos sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre as diferentes entradas conforme descrito a seguir:

- O comportamento do BubbleSort clássico sobre os quatro cenários de entrada;
- O comportamento do BubbleSort otimizados sobre os quatro cenários de entrada;
- O comportamento do BubbleSort tradicional frente sua versão otimizada;
- O comportamento do SelectionSort tradicional frente sua versão otimizada.



PROPOSTA 2 – Igor França, Mateus Karvat e Amanda Graeff

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação aos métodos de ordenação quadráticos, bem como ilustrar o comportamento apresentados pelas estratégias em diferentes cenários.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) BubbleSort;
- b) InsertionSort;
- c) SelectionSort.

Deve-se avaliar o comportamento dos métodos perante conjuntos de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TnsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os três métodos de ordenação trabalhando sobre conjunto aleatórios;
- Os três métodos de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todos os métodos sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- BubbleSort sobre os quatro tipos de entrada;
- SelectionSort sobre os quatro tipos de entrada;
- InsertionSort sobre os quatro tipos de entrada;
- Análise entre o desempenho de todos os métodos sobre o conjunto de entradas aleatórias;

- Análise entre o desempenho de todos os métodos sobre o conjunto de entradas ordenadas;
- Análise entre o desempenho de todos os métodos sobre o conjunto de entradas parcialmente ordenadas;
- Análise entre o desempenho de todos os métodos sobre o conjunto de entradas decrescentes.



PROPOSTA 3 – Angelo Orssatto e Gabriel Girardi da Costa

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação aos métodos de ordenação de ordem $O(n \log_2 n)$, bem como ilustrar o comportamento apresentados pelas estratégias em diferentes cenários.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) MergeSort;
- b) Quicksort;
- c) HeapSort.

Deve-se avaliar o comportamento dos métodos perante conjuntos de testes com diferentes características. Durante a execução dos experimentos o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pela dupla.
- A forma com que o pivô é escolhido para o Quicksort é de escolha do grupo. Recomenda-se não usar o primeiro elemento do vetor como pivô.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TNsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os três métodos de ordenação trabalhando sobre conjunto aleatórios;
- Os três métodos de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todos os métodos sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- Quicksort sobre os quatro tipos de entrada;
- MergeSort sobre os quatro tipos de entrada;
- HeapSort sobre os quatro tipos de entrada;

- Análise do desempenho dos três métodos sobre o conjunto de dados ordenado;
- Análise do desempenho dos três métodos sobre o conjunto de dados parcialmente ordenado;
- Análise do desempenho dos três métodos sobre o conjunto de dados aleatórios;
- Análise do desempenho dos três métodos sobre o conjunto de dados decrescentes.



PROPOSTA 4 – Ana Paula Bernardes e Willian Faino

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação MergeSort, bem elucidar se a passagem por referência, trabalhando com um único vetor em memória é mais vantajosa que criar uma cópia do vetor a cada chamada recursiva do algoritmo.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- Abordagem clássica do MergeSort que aloca um vetor temporário a cada chamada recursiva (solução vista em aula);
- Abordagem alternativa do MergeSort que, ao invés de instanciar um novo vetor a cada chamada, simplesmente sobrepõe o vetor de entrada (que foi passado por referência).

Deve-se avaliar o comportamento dos métodos perante conjuntos de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pela dupla.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TnsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- As duas estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- Comparação do MergeSort clássico trabalhando sobre todos os tipos de entrada;
- Comparação do MergeSort que usa a passagem por referência trabalhando sobre todos os tipos de entrada;

- Análise comparativa entre as duas abordagens do merge para os quatro cenários de entrada dos dados.



PROPOSTA 5 – Rodrigo Campos e Valquíria Belusso

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação BucketSort, bem como ilustrar o impacto do uso de diferentes números de baldes na aplicação da técnica.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) InsertionSort tradicional
- b) Abordagem alternativa do InsertionSort usando a estratégia de BucketSort, ou seja, o conjunto de dados dos vetores de entrada devem ser particionados em baldes e então ordenados pelo método de InsertionSort.
 1. Para tanto, deverão ser testadas quatro abordagens distintas da técnica implementada, cada uma com uma quantidade diferente de baldes, a saber:
 - i. 10 baldes
 - ii. 50 baldes
 - iii. 100 baldes
 - iv. 1000 baldes

Deve-se avaliar o comportamento dos métodos perante conjuntos de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- Quando o número de elementos no vetor for menor do que o número de baldes não há necessidade de análise. Dessa forma, alguns gráficos serão parciais, ou seja, não terá análise de elementos menores. Por exemplo, ao se analisar o método com 1000 baldes, não serão testados os conjuntos de entrada compostos por 100, 200 e 500 elementos.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TnsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os cinco métodos (Insertion tradicional e cada número de baldes) de ordenação trabalhando sobre conjunto aleatórios;

- As cinco estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- Usar 10 baldes sobre os quatro tipos de entrada;
- Usar 50 baldes sobre os quatro tipos de entrada;
- Impacto de usar 100 baldes sobre os quatro conjuntos de entrada;
- Influência de se usar 1000 baldes nos conjuntos de entrada;
- Comparar o desempenho do InsertionSort tradicional frente ao melhor desempenho ao se usar o BucketSort para os quatro cenários de entrada.



PROPOSTA 6 – Lucas Mülling

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação Quicksort em comparação a um método de ordenação de ordem linear.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) Quicksort;
- b) Countingsort.

Deve-se avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de sua escolha.
- A forma com que os métodos serão implementados é determinada pelo(a) aluno(a).
- Para a escolha do pivô do Quicksort devem ser sorteados três elementos aleatórios do conjunto e escolhida a mediana deste conjunto como pivô para aquela etapa do método. Se o vetor tiver menos que 3 elementos, usar um elemento escolhido aleatoriamente.
- Durante a execução do Countingsort para se descobrir qual o maior valor presente no conjunto de entrada deve-se fazer uma varredura no conjunto. Essa varredura deve entrar na contagem de tempo do método.

- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:

Valores Aleatórios

https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r

Valores Decrescentes

<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>

Valores Ordenados Crescentes

https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TNsneXsK

Valores Parcialmente Ordenados

<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- As duas estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- Quicksort sobre os quatro tipos de entrada;
- Countingsort sobre os quatro tipos de entrada;
- Análise do desempenho do Countingsort frente ao Quicksort para os quatro tipos de entrada, identificando assim, qual método foi melhor em cada cenário.



PROPOSTA 7 – João Dotta, Gabriel Andrade de Araújo e Alonso Fritz

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação Quicksort analisando-se seu desempenho quando implementado de forma recursiva frente à uma implementação iterativa.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) Quicksort recursivo;
- b) Quicksort iterativo.

Deve-se avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- Para a determinação do pivô deve-se escolher o elemento central do vetor.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TnsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- As duas estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- Quicksort recursivo sobre os quatro tipos de entrada;
- Quicksort iterativo sobre os quatro tipos de entrada;
- Análise do desempenho da solução recursiva frente a iterativa sobre os quatro conjuntos de entrada, identificando qual solução foi mais eficiente em cada um dos cenários.

PROPOSTA 8 – Pedro Xavier e Vitor Gilnek

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação MergeSort analisando-se seu desempenho quando implementado de forma a dividir o problema em dois ou quatro subproblemas.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) MergeSort clássico (dois subproblemas);
- b) Super MergeSort (quatro subproblemas).

Deve-se avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pela dupla.
- Em casos em que há menos de 4 elementos no subproblema, pode-se aplicar a divisão em dois subconjuntos.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TNsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- As duas estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- MergeSort clássico sobre os quatro tipos de entrada;
- Super MergeSort sobre os quatro tipos de entrada;

- Análise do desempenho da solução Super MergeSort frente a solução clássica sobre os quatro conjuntos de entrada, identificando qual solução foi mais eficiente em cada um dos cenários.



PROPOSTA 9 – Igor Engler e Marcos Mucelini

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação InsertionSort analisando-se seu desempenho frente à sua evolução, o método ShellSort.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) InsertionSort;
- b) ShellSort (Knuth, 1973).
 - a. Neste caso a sequência seria $h=1, 4, 13, 40, 121, 364, 1093, \dots$

Deve-se avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pela dupla.
- A escolha dos valores de h deve se dar da seguinte forma:

$$h(s) = 1 \text{ para } s = 1$$
$$h(s) = \frac{(3^h - 1)}{2} \text{ para } s > 1$$

- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TNsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- As duas estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- O comportamento do InsertionSort sobre os quatro tipos de entrada;
- O ShellSort sobre os quatro tipos de entrada;
- O comportamento do ShellSort frente ao InsertionSort em todos os tipos de entrada, identificado que método foi mais eficiente em cada um dos quatro cenários.



PROPOSTA 10 – Gabriel Caetano Fermino, Taylor Byk Schinvelski e Igor Kaiser Gris

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação ShellSort, entendendo o impacto da escolha do tamanho dos segmentos do processo.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) ShellSort (Shell, 1959);
 - a. Neste caso a sequência seria $h=1, 2, 5, 10, 21, \dots$
- b) ShellSort (Knuth, 1973).
 - a. Neste caso a sequência seria $h=1, 4, 13, 40, \dots$

Deve-se avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- A escolha dos valores de h deve se dar da seguinte forma:

$$\begin{cases} h(s) = 1 & s = 1 \\ h(s) = \left\lfloor \frac{h}{2^s} \right\rfloor & s > 1 \end{cases}$$
$$h(s) = 1 \text{ para } s = 1$$
$$h(s) = \frac{(3^h - 1)}{2} \text{ para } s > 1$$

- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TnsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;

- As duas estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- O comportamento do ShellSort (Shell, 1953) sobre os quatro tipos de entrada;
- O ShellSort (Knuth, 1973) sobre os quatro tipos de entrada;
- O desempenho da implementação proposta por Shell frente a proposta por Knuth em cada um dos quatro tipos de entrada, identificando quem apresentou o melhor desempenho em cada um dos quatro cenários.



PROPOSTA 11 – Maurício Hiraki Ishida

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação ao método de ordenação Quicksort analisando-se seu desempenho quando implementado a escolha do pivô com base em critérios distintos.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- Quicksort com pivô escolhido na posição central do vetor a ser ordenado;
- Quicksort com pivô escolhido a partir da mediada de 5 elementos aleatoriamente definidos. Se o vetor for composto por menos que 5 elementos, pode-se sortear apenas três para se obter a mediana e, caso o tamanho do vetor seja menor que 3, apenas sortear um dos elementos como pivô;
- Quicksort em que a escolha do pivô sempre pega o primeiro elemento do vetor a ser ordenado.

Deve-se avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características. Durante a execução dos testes o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis:
 - Valores Aleatórios
https://drive.google.com/open?id=1ftGLbzSoUHS5_8iy_IUWEu5Gu--_kR4r
 - Valores Decrescentes
<https://drive.google.com/open?id=1AepNYTK6WxEhmL8D2TtaTtaWR0KvsuYZ>
 - Valores Ordenados Crescentes
https://drive.google.com/open?id=1fGHIMiOAEhlOS8_-4Lh7E0b1TNsneXsK
 - Valores Parcialmente Ordenados
<https://drive.google.com/file/d/1BsQFtumOmDCBeqcc66U-V2NoOlpFM2Wt/view?usp=sharing>

Devem ser construídos quatro conjuntos de testes, conforme os arquivos disponíveis nos links apresentados acima:

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- As duas estratégias de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todas as abordagens sobre conjuntos parcialmente ordenados de forma crescente.


Além disso, deve ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- As três variações do Quicksort sobre os quatro tipos de entrada;

- Analisar o comportamento das três variações sobre cada um dos cenários comparando seus desempenhos a fim de identificar em que cenário cada método foi melhor.

O que deve ser entregue:

Relatório em formato pdf contendo:

- Descrição de como foi realizado o processo empírico de determinação dos custos: cenário de realização dos experimentos e como foram tomadas as métricas exigidas.
 - Detalhar a configuração usada nos testes (Processador, SO, IDE, etc..).
- Gráficos evidenciando o comportamento dos métodos perante todos os cenários considerando o tamanho dos conjuntos de entrada.
-  ○ Análise do comportamento dos métodos durante a execução dos testes.
 - Esta análise deve ser feita com bastante critério e ser esclarecedora, apontando razões para os comportamentos observados.

Códigos fonte usados na execução dos experimentos

- Não é necessário o envio dos projetos, apenas os códigos fonte
- Manter no código os comandos usados para tomada de tempo

Quando:

A data de entrega está marcada para dia 07/03/2022 até as 23:59.

A entrega do trabalho deve ser feita via Teams.

No dia 08/03/2022 haverá então a apresentação das soluções desenvolvidas em ordem de proposta, de 1 a 11, iniciando às 08:00.

A qualidade da apresentação também constitui critério de avaliação.