

Trabalho individual 2 - Valor 10 pontos

INFORMAÇÕES DOCENTE						
CURSO: ENGENHARIA DE SOFTWARE	DISCIPLINA: FUNDAMENTOS DE PROJETO E ANÁLISE DE ALGORITMOS	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA: 5º
					x	
PROFESSOR (A): João Paulo Carneiro Aramuni						

Enunciado do projeto: Implementação do Algoritmo de Seleção Simultânea do Maior e do Menor Elementos (MaxMin Select) em Python

Objetivo:

- Desenvolver um programa em Python que implemente o algoritmo de seleção simultânea do maior e do menor elementos (MaxMin Select) de uma sequência de números, utilizando a abordagem de divisão e conquista. O projeto deverá ser entregue por meio de um link para o repositório do GitHub no CANVAS.

Sobre o algoritmo:

- O algoritmo de seleção simultânea (MaxMin Select) pode ser implementado de forma recursiva, utilizando a técnica de divisão e conquista. O problema é dividido em subproblemas menores que são resolvidos recursivamente, e seus resultados são combinados para encontrar o maior e o menor elementos com eficiência. Esse método reduz o número de comparações necessárias em comparação com uma abordagem ingênua.

Requisitos do projeto:

1. Código Python:

- O programa deverá conter a implementação do algoritmo MaxMin Select em um arquivo chamado main.py.

2. Documentação no README.md:

- O repositório deverá incluir um arquivo README.md que explique como rodar o projeto e também a lógica do algoritmo implementado.
- O README deverá ser estruturado conforme o exemplo fornecido neste *repo*:
<https://github.com/joaopauloaramuni/fundamentos-de-projeto-e-analise-de-algoritmos/tree/main/PROJETOS>

- O README deverá conter:
 - Descrição do projeto: Explicação do algoritmo e da lógica de como ele foi implementado (linha a linha).
 - Como executar o projeto: Instruções para rodar o código no ambiente local.
 - Relatório técnico contendo as análises sobre o algoritmo.

3. Relatório técnico incorporado ao README:

- Análise da complexidade assintótica pelo método de contagem de operações:
 - Explique detalhadamente o número de comparações realizadas em cada etapa do algoritmo, considerando a divisão do problema em subproblemas e a combinação dos resultados. Calcule o total de comparações realizadas para n elementos e mostre como isso resulta em uma complexidade temporal $O(n)$.
 - Exemplos:
 - AULA 01_Análise de complexidade de algoritmos.pdf
 - <https://github.com/joaopauloaramuni/fundamentos-de-projeto-e-analise-de-algoritmos/tree/main/PDF>
- Análise da complexidade assintótica pela aplicação do Teorema Mestre:
 - Considere a recorrência do MaxMin Select:
 - $T(n) = 2T(n/2) + O(1)$
 - Perguntas:
 - 1. Identifique os valores de a , b e $f(n)$ na fórmula:
 - $T(n) = a \cdot T(n/b) + f(n)$.
 - 2. Calcule $\log_b a$ para determinar o valor de p .
 - 3. Determine em qual dos três casos do Teorema Mestre esta recorrência se enquadra.
 - 4. Encontre a solução assintótica ($T(n)$) da fórmula.
 - Exemplos:
 - AULA 01_Análise de complexidade de algoritmos.pdf
 - <https://github.com/joaopauloaramuni/fundamentos-de-projeto-e-analise-de-algoritmos/tree/main/PDF>

4. Entrega:

- O projeto deverá ser enviado por meio de um repositório no GitHub, com o link postado no sistema CANVAS. Certifique-se de que o repositório esteja público ou acessível (antes de realizar a entrega, faça um teste em uma aba anônima do navegador).
- Exemplo de link a ser entregue no CANVAS:
 - https://github.com/exemploaluno/trabalho_individual_2_FPAA

Critérios de avaliação:

1. Implementação do algoritmo (50%):

- O código está correto e eficiente?
- A lógica do algoritmo MaxMin Select foi seguida adequadamente?

2. Documentação no README.md (50%):

- O README segue o padrão especificado?
- O relatório técnico está claro e apresenta uma análise detalhada?
- A análise da complexidade assintótica, em ambos os métodos, está correta e bem explicada?

Dicas para o desenvolvimento:

- Comece com a implementação básica do algoritmo MaxMin Select recursivo.
- Teste o algoritmo com diferentes tamanhos e conteúdos de sequência para garantir sua precisão.
- Consulte o material e leia a aula sobre complexidade assintótica e Teorema Mestre para enriquecer o relatório.
 - AULA 01_Análise de complexidade de algoritmos.pdf
 - <https://github.com/joaopauloaramuni/fundamentos-de-projeto-e-analise-de-algoritmos/tree/main/PDF>

Ponto extra:

- Crie um diagrama visual para ilustrar a divisão e combinação no algoritmo MaxMin Select:
 - Use ferramentas gráficas como Lucidchart, draw.io, ou PowerPoint para desenhar a estrutura hierárquica da recursão no algoritmo.
 - O diagrama deve incluir:
 - 1. Como o problema original é dividido em subproblemas menores (representando as chamadas recursivas).
 - 2. Como os subproblemas são combinados para obter o resultado final (os maiores e menores elementos).
 - 3. Os níveis da árvore de recursão e o número de comparações realizadas em cada nível.
 - Salve o diagrama em formato PNG e adicione-o ao repositório no GitHub, em uma pasta chamada assets.
 - Inclua uma referência ao diagrama no README.md para contextualizar sua lógica.
-