

**Trabalho individual 1 - Valor 5 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 Karatsuba em Python**

**Objetivo:**

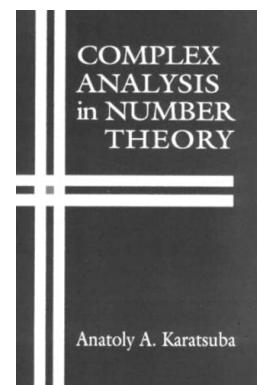
- Desenvolver um programa em Python que implemente o algoritmo de Karatsuba para multiplicação eficiente de dois números inteiros. O projeto deverá ser entregue por meio de um link para o repositório do GitHub no CANVAS.

**Sobre o algoritmo:**

- O algoritmo de Karatsuba é uma técnica eficiente para multiplicação de números inteiros grandes, introduzida por Anatolii Karatsuba em 1960. Ele melhora a complexidade da multiplicação em comparação ao método tradicional de multiplicação direta.



*Anatoli Alexeievitch Karatsuba*



22/11/1994

**Requisitos do projeto:**

**1. Código Python:**

- O programa deverá conter a implementação do algoritmo de Karatsuba 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.



## PUC Minas

- 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 ciclomática**:
  - Representar o fluxo de controle da função.
  - Estruturar o grafo de fluxo (nós e arestas).
  - Calcular a complexidade ciclomática usando a fórmula  $M = E - N + 2P$ , onde:
    - $E$ : Número de arestas no grafo.
    - $N$ : Número de nós no grafo.
    - $P$ : Número de componentes conexos no grafo (em um único programa,  $P = 1$ ).
- Análise da **complexidade assintótica**:
  - Determinar a complexidade temporal e espacial do algoritmo.
  - Identificar os três casos principais: melhor caso, caso médio e pior caso.

### 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).

### Critérios de avaliação:

#### 1. Implementação do algoritmo (50%):

- O código está correto e eficiente?
- A lógica do algoritmo de Karatsuba 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 ciclomática e assintótica está correta e bem explicada?

Dicas para o desenvolvimento:

- Comece com a implementação básica do algoritmo de Karatsuba.
- Teste o algoritmo com diferentes pares de números para garantir sua precisão.
- Consulte o material e leia a aula sobre complexidade ciclômica e assintótica para enriquecer o relatório.
  - AULA 00\_Revisão e nivelamento.pdf
  - <https://github.com/joaopauloaramuni/fundamentos-de-projeto-e-analise-de-algoritmos/tree/main/PDF>

Ponto extra:

Use ferramentas gráficas (como o Diagrama de Fluxo no Lucidchart ou draw.io) para desenhar o grafo de fluxo que represente a estruturação de nós e arestas da análise ciclômica.

Exemplo de link a ser entregue no CANVAS:

[https://github.com/exemploaluno/trabalho\\_individual\\_1\\_FPAA](https://github.com/exemploaluno/trabalho_individual_1_FPAA)

---