

# **Relatório sobre o Paradigma Programação Gulosa para o problema de Apagando e Ganhando em Python**

**Nome: Daniel Antônio de Sá**

**Disciplina: PAA**

Este trabalho tem como objetivo apresentar um solucionador eficiente para o problema "Apagando e Ganhando" e explicar como as técnicas de programação gulosa são aplicadas para resolver esse desafio..

O problema "Apagando e Ganhando" envolve uma entrada composta por N dígitos de um número inteiro e D dígitos que devem ser removidos. O objetivo é remover os dígitos de forma a gerar o maior número possível, sem alterar a sequência da entrada. A seguir, são fornecidos exemplos do problema e suas soluções correspondentes:

Exemplo de Entrada	Exemplo de Saída
--------------------	------------------

4 2	
-----	--

3759	79
------	----

6 3	
-----	--

123123	323
--------	-----

Para resolver o problema "Apagando e Ganhando", optamos por adotar a abordagem da programação gulosa. Um algoritmo guloso constrói a solução tomando decisões que parecem ser as melhores localmente, na esperança de que essas escolhas levem à solução ideal globalmente. Isso significa que o algoritmo é decisivo em suas escolhas, não volta atrás em decisões tomadas e constrói a solução de forma direta. A eficácia das soluções gulosas depende de duas características fundamentais:

**Subestruturas Ótimas:** A solução ideal para o problema contém soluções ótimas para seus subproblemas.

**Característica Gulosa:** Ao fazer uma escolha que parece ser a melhor localmente e continuar resolvendo o problema, é possível chegar à solução ideal sem a necessidade de reconsiderar decisões anteriores.

No contexto do problema "Apagando e Ganhando", a abordagem de programação gulosa se mostra altamente adequada. Nosso algoritmo é construído dentro de um loop de repetição que continua até que os valores de N e D sejam iguais a zero. Uma vez obtidos esses valores e o número que será avaliado, o algoritmo percorre o número a partir do dígito de maior peso até o dígito de menor peso,

utilizando um array auxiliar para armazenar os dígitos. A cada iteração, verifica-se se o dígito atual é maior do que os últimos dígitos adicionados. Se for maior, remove-se os dígitos anteriores, garantindo que não ultrapassem a quantidade de dígitos que devem ser removidos. Além disso, há um terceiro loop para remover os dígitos restantes, caso estejam em ordem, mas a quantidade desejada ainda não tenha sido atingida. Essa etapa começa a partir do dígito de menor peso e, finalmente, gera o valor final.

Neste trabalho, apresentamos uma solução eficiente para o problema "Apagando e Ganhando" utilizando a abordagem da programação gulosa. Demonstramos como essa técnica, que se baseia em tomar decisões locais aparentemente melhores, pode levar a uma solução global ideal. A solução proposta utiliza iterações e verificações cuidadosas para garantir que os dígitos sejam removidos de forma a gerar o maior número possível, sem alterar a sequência original. A aplicação bem-sucedida da programação gulosa no contexto desse problema ilustra a importância de escolher a estratégia certa para abordar desafios algorítmicos complexos.