TRABALHO PAA - ALGORITMOS

Nome: Guilherme Silva Cardoso RA: 840122

Nome: Marcos Antonio Carlos Garcia RA: 840121

Link do video: https://youtu.be/NG8a-ey7Y30

```python
import sys

def desconto(n, precos):
    # Ordena os preços em ordem decrescente
    precos.sort(reverse=True)
    desconto = 0

    # A cada conjunto de 3 produtos, pegamos o menor (terceiro da sequência ordenada)
    for i in range(2, n, 3):
        desconto += precos[i]

    return desconto

n = int(input().strip())
precos = list(map(int, input().strip().split()))
print(desconto(n, precos))
```

Explicação:

Ordenação: Os preços são ordenados em ordem decrescente para garantir que os produtos mais caros venham primeiro.

Seleção do desconto: A cada grupo de três produtos consecutivos, o terceiro (o mais barato do grupo) é escolhido para o desconto.

Soma dos descontos: Os preços selecionados no passo anterior são somados para calcular o valor total do desconto.

Por que é guloso?

Sim, porque faz a melhor escolha local em cada passo (pega o menor dos três para maximizar o desconto), esperando que isso leve a um resultado ótimo global.

```python
import math

def exponentiation(base, exp):
    if exp == 0:
        return 1
    elif exp % 2 == 0:
        half = exponentiation(base, exp // 2)
        return half * half
    else:
        return base * exponentiation(base, exp - 1)

def exponential(n1, n2):
    result = exponentiation(n1, n2)
    ret1 = int(str(result)[0])
    ret2 = int(math.floor(math.log10(result))) if result != 0 else 0
    return ret1, ret2

if __name__ == "__main__":
    n1, n2 = map(float, input().strip().split())
    n1 = float(n1)
    n2 = int(n2)
    ret1, ret2 = exponential(n1, n2)
    print(f"{ret1} {ret2}")
```

Divisão e Conquista: Exponenciação

Objetivo: Calcular (base^exp) de forma rápida. Como funciona:

Se o expoente for 0, o resultado é 1.

Se o expoente for par, divide o expoente por 2 e calcula a potência de forma recursiva, depois eleva ao quadrado.

Se o expoente for ímpar, diminui o expoente em 1, calcula a potência recursivamente e multiplica pela base.

Eficiência: Ao dividir o expoente pela metade a cada passo, a complexidade diminui de O(n) para O(log n), tornando o algoritmo muito mais rápido.

Essa é a essência da divisão e conquista: dividir o problema em subproblemas menores e combiná-los para obter a solução final de forma eficiente.
