



Universidad Nacional Autónoma de México



Facultad de Estudios Superiores Aragón

Ingeniería en
Computación

Diseño y análisis de
algoritmos

TAREA 6

Alumno: Velazco Martínez Brayan

Profesor: Hernández Cabrera
Jesús

¿Por que elegí este problema?

Elegí este problema porque me resulta más sencillo comprender la técnica de Divide y Vencerás mediante este tipo de ejercicios, ya que permite visualizar claramente cómo se descompone un problema en partes más pequeñas y luego se combinan los resultados. Además, trabajar con números positivos y negativos aumenta la complejidad del análisis, lo que me ayuda a desarrollar mi razonamiento lógico y fortalecer mi pensamiento analítico. Este problema también me permitió aplicar la recursión de una manera práctica, comprendiendo mejor cada una de las etapas del método: división, resolución y combinación. Por todo ello, considero que es una opción adecuada para afianzar mis conocimientos sobre el paradigma Divide y Vencerás y su aplicación en la programación.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes standard menu items: File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled "Tarea 6". On the left, there's a sidebar with icons for Explorer, Search, Find, and others, followed by a tree view under "TAREA 6" showing a file named "productoMaximo.py". The main area displays the Python code for calculating the maximum product of a subarray. The code uses two recursive functions, `producto_maximo` and `producto_cruzado`, and a main block to print the result for a specific array. The bottom navigation bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is active), and PORTS. The terminal tab shows the command run and the output "El producto máximo es: 40". The status bar at the bottom right indicates the current file is "productoMaximo.py", with details like "Ln 10, Col 31", "Spaces: 4", "UTF-8", "CRLF", "Python", and "3.12.3".

```
def producto_maximo(arr, inicio, fin):
    max_izq = producto_maximo(arr, inicio, medio)
    max_der = producto_maximo(arr, medio + 1, fin)

    max_cruzado = producto_cruzado(arr, inicio, medio, fin)

    return max(max_izq, max_der, max_cruzado)

def producto_cruzado(arr, inicio, medio, fin):
    prod_temp = 1
    mejor_izq = float('-inf')
    for i in range(medio, inicio - 1, -1):
        prod_temp *= arr[i]
        if prod_temp > mejor_izq:
            mejor_izq = prod_temp

    prod_temp = 1
    mejor_der = float('-inf')
    for j in range(medio + 1, fin + 1):
        prod_temp *= arr[j]
        if prod_temp > mejor_der:
            mejor_der = prod_temp

    return mejor_izq * mejor_der

if __name__ == "__main__":
    arreglo = [9, 1, 4, -1, 0, 1, 5, 8]
    resultado = producto_maximo(arreglo, 0, len(arreglo) - 1)
    print("El producto máximo es:", resultado)
```