



TOLENTINO ROMERO GERARDO



Analisis y Diseño de Algoritmos

Profesor. Andres García Floriano

Practica 5

Utilizamos un enfoque de **dos punteros** para resolver el problema de manera eficiente en $O(n)O(n)O(n)$:

1. Colocamos un puntero al inicio y otro al final del arreglo.
2. Calculamos el área entre las líneas donde se encuentran los punteros y la guardamos si es la mayor encontrada hasta el momento.
3. Movemos el puntero de la línea de menor altura para intentar maximizar el área.
4. Repetimos el proceso hasta que los punteros se encuentren.

Este enfoque permite optimizar el tiempo de ejecución y encontrar la solución sin recorrer todas las posibles combinaciones de líneas.

```
def max_area(altura):
    izquierda = 0
    derecha = len(altura) - 1
    max_area = 0

    while izquierda < derecha:
        # Calcula el área actual
        area_actual = min(altura[izquierda], altura[derecha]) * (derecha - izquierda)
        # Actualiza el área máxima si se encuentra una mayor
        max_area = max(max_area, area_actual)

        # Mueve el puntero de la línea de menor altura
        if altura[izquierda] < altura[derecha]:
            izquierda += 1
        else:
            derecha -= 1

    return max_area
```

Para el siguiente arreglo de alturas: altura = [3, 1, 4, 7, 2, 5, 6, 8, 3, 9, 4, 6], los resultados son los siguientes.

La mayor cantidad de liquido que se puede almacenar es: 48