_4bucles/_10optimizacionsubarreglo.py

```python
# Ejercicio 10: Problema de Optimización de Subarreglo:

# Imagina que estás trabajando en un sistema de análisis de datos y te han proporcionado una lista de números enteros. Tu tarea es desarrollar una función llamada max_subarray_sum que encuentre y devuelva la suma máxima de un subarreglo contiguo en la lista.

# Por ejemplo, considera la lista [1, -2, 3, 10, -4, 7, 2, -5]. El subarreglo contiguo con la suma máxima es [3, 10, -4, 7, 2], y la suma de esos elementos es 18. Por lo tanto, la función debería devolver 18 para esta lista.

# Implementa la función max_subarray_sum y, además, aplica memoización para mejorar su eficiencia en el cálculo de subarreglos de suma máxima en listas previamente procesadas.
import functools

@functools.lru_cache(maxsize=None)
def max_subarray_sum(arr):
  if not arr:
    return 0

  current_sum=max_sum=arr[0]
  max_subarray=[arr[0]]

  for num in arr[1:]:
    if num >current_sum+num:
      current_sum=num
      max_subarray=[num]
    else:
      current_sum+=num
      max_subarray.append(num)
    max_sum=max(max_sum,current_sum)
  return max_sum,max_subarray

arreglo = tuple([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
resultado,subarray=max_subarray_sum(arreglo)
print("Suma maxima del subarreglo contiguo es de: ",resultado)
print("SUbarreglo contiguo con la suma maxima es: ",subarray)

```