In [1]:

```
import time
import functools
```

In [2]:

```
def timing_and_logging(log_to_file=False, file_name='function_logs.txt'):
    def decorator(func):
        @functools.wraps(func)
        def wrapper(*args, **kwargs):
            start_time = time.time()
            start_message = f"Starting '{func.__name__}' at {time.ctime(start_time)}"


            if log_to_file:
                with open(file_name, 'a') as f:
                    f.write(start_message + '\n')
            else:
                print(start_message)


            result = func(*args, **kwargs)

            end_time = time.time()
            end_message = f"Finished '{func.__name__}' at {time.ctime(end_time)}"
            execution_time_message = f"Execution time: {end_time - start_time:.4f} secon


            if log_to_file:
                with open(file_name, 'a') as f:
                    f.write(end_message + '\n')
                    f.write(execution_time_message + '\n\n')
            else:
                print(end_message)
                print(execution_time_message)

            return result
        return wrapper
    return decorator
```

In [3]:

```
@timing_and_logging(log_to_file=False)
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)
```

In [4]:

```
@timing_and_logging(log_to_file=True)
def cpu_intensive_calculation(n):
    result = 0
    for i in range(n):
        result += i ** 2
    return result
```

In [5]:

```
sorted_array = quicksort([3, 6, 8, 10, 1, 2, 1])
```

```
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Starting 'quicksort' at Sun Nov 10 22:30:20 2024
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
Finished 'quicksort' at Sun Nov 10 22:30:20 2024
Execution time: 0.0000 seconds
```

In [6]:

```
result = cpu_intensive_calculation(100000)
```

In [7]:

```
with open('function_logs.txt', 'r') as f:
    print(f.read())
```

```
Starting 'cpu_intensive_calculation' at Sun Nov 10 22:30:32 2024
Finished 'cpu_intensive_calculation' at Sun Nov 10 22:30:32 2024
Execution time: 0.0216 seconds
```

In [ ]: