Assignment – 2.3

NAME : s.harshavardhan

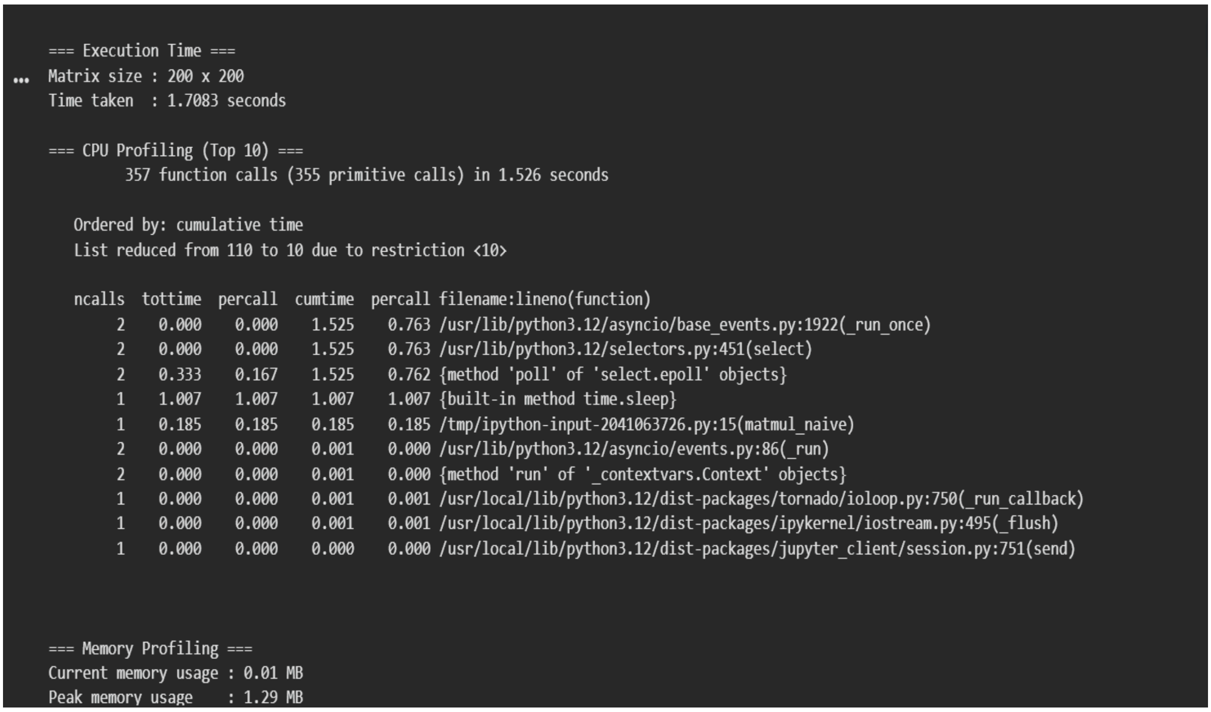
HT. NO : 2303A51563

BATCH : 10

CODE:

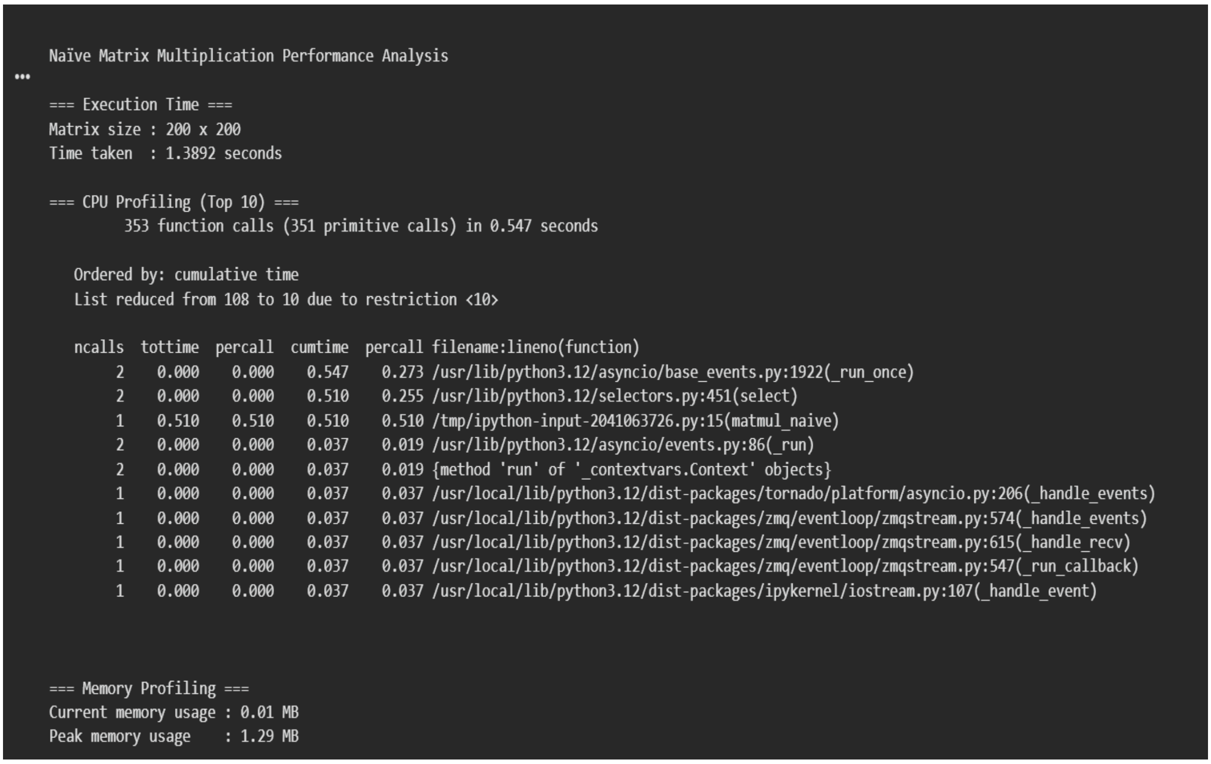
```
import random
import time
import cProfile
import pstats
import io
import tracemalloc
# ------------------------------------------------
# Generate a random n x n matrix
# ------------------------------------------------
def generate_matrix(n):
    return [[random.random() for _ in range(n)] for _ in range(n)]
# ------------------------------------------------
# Naïve matrix multiplication (O(n^3))
# ------------------------------------------------
def matmul_naive(A, B):
    n = len(A)
    C = [[0.0 for _ in range(n)] for _ in range(n)]
    for i in range(n):
        for j in range(n):
            for k in range(n):
                C[i][j] += A[i][k] * B[k][j]
    return C
# ------------------------------------------------
# Measure execution time
# ------------------------------------------------
def run_timing(n):
    A = generate_matrix(n)
    B = generate_matrix(n)

    start = time.perf_counter()
    matmul_naive(A, B)
    end = time.perf_counter()
    print("=== Execution Time ===")
    print(f"Matrix size : {n} x {n}")
    print(f"Time taken  : {end - start:.4f} seconds\n")
# ------------------------------------------------
# CPU profiling using cProfile
# ------------------------------------------------
def run_cpu_profile(n):
    A = generate_matrix(n)
    B = generate_matrix(n)
    profiler = cProfile.Profile()
    profiler.enable()
    matmul_naive(A, B)
    profiler.disable()
    s = io.StringIO()
    stats = pstats.Stats(profiler, stream=s).sort_stats("cumulative")
    stats.print_stats(10)
    print("=== CPU Profiling (Top 10) ===")
    print(s.getvalue())

# ------------------------------------------------
# Memory profiling using tracemalloc
# ------------------------------------------------
def run_memory_profile(n):
    A = generate_matrix(n)
    B = generate_matrix(n)
    tracemalloc.start()
    matmul_naive(A, B)
    current, peak = tracemalloc.get_traced_memory()
    tracemalloc.stop()
    print("=== Memory Profiling ===")
    print(f"Current memory usage : {current / 10**6:.2f} MB")
    print(f"Peak memory usage    : {peak / 10**6:.2f} MB\n")
# ------------------------------------------------
# Main function
# ------------------------------------------------
def main():
    N = 200   # Change carefully: 100, 200, 300 (higher = much slower)
    print("\nNaïve Matrix Multiplication Performance Analysis\n")
    run_timing(N)
    run_cpu_profile(N)
    run_memory_profile(N)

if __name__ == "__main__":
    main()
```

GoogleColab - CPU :



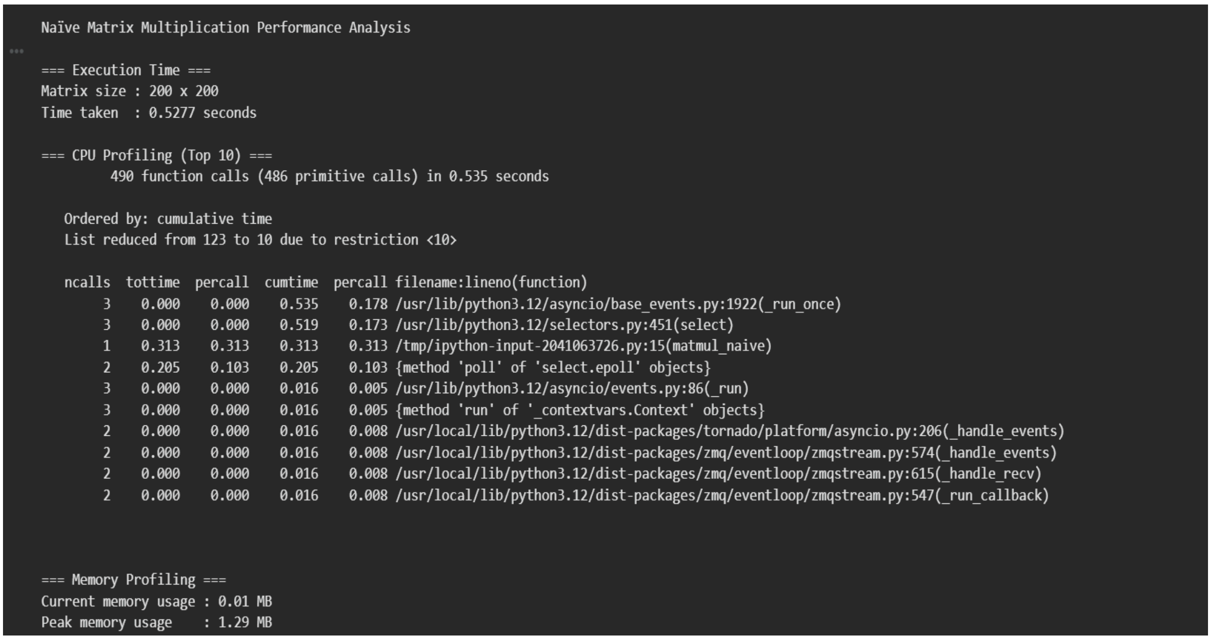
```
=== Execution Time ===
Matrix size : 200 x 200
Time taken  : 1.7083 seconds

=== CPU Profiling (Top 10) ===
         357 function calls (355 primitive calls) in 1.526 seconds

   Ordered by: cumulative time
   List reduced from 110 to 10 due to restriction <10>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        2    0.000    0.000    1.525    0.763 /usr/lib/python3.12/asyncio/base_events.py:1922(_run_once)
        2    0.000    0.000    1.525    0.763 /usr/lib/python3.12/selectors.py:451(select)
        2    0.333    0.167    1.525    0.762 {method 'poll' of 'select.epoll' objects}
        1    1.007    1.007    1.007    1.007 {built-in method time.sleep}
        1    0.185    0.185    0.185    0.185 /tmp/ipython-input-2041063726.py:15(matmul_naive)
        2    0.000    0.000    0.001    0.000 /usr/lib/python3.12/asyncio/events.py:86(_run)
        2    0.000    0.000    0.001    0.000 {method 'run' of '_contextvars.Context' objects}
        1    0.000    0.000    0.001    0.001 /usr/local/lib/python3.12/dist-packages/tornado/ioloop.py:750(_run_callback)
        1    0.000    0.000    0.001    0.001 /usr/local/lib/python3.12/dist-packages/ipykernel/iostream.py:495(_flush)
        1    0.000    0.000    0.000    0.000 /usr/local/lib/python3.12/dist-packages/jupyter_client/session.py:751(send)



=== Memory Profiling ===
Current memory usage : 0.01 MB
Peak memory usage    : 1.29 MB
```

GoogleColab – GPU :



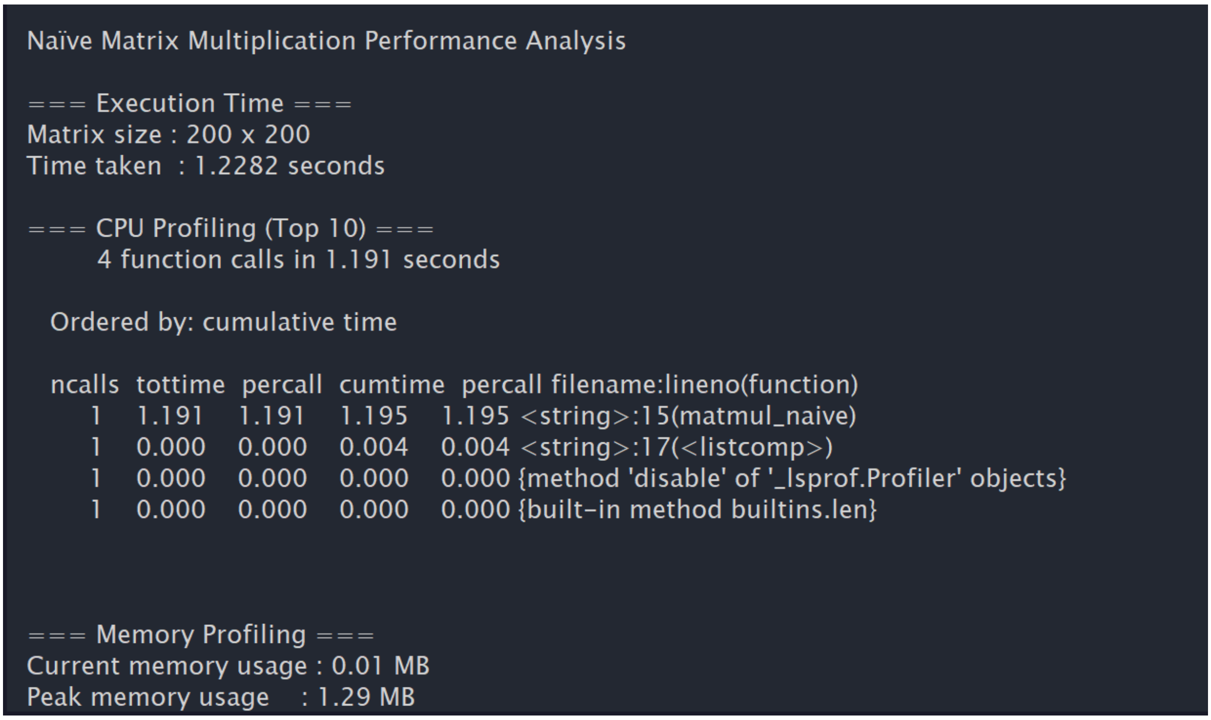
```
Naïve Matrix Multiplication Performance Analysis
•••
=== Execution Time ===
Matrix size : 200 x 200
Time taken  : 1.3892 seconds

=== CPU Profiling (Top 10) ===
         353 function calls (351 primitive calls) in 0.547 seconds

   Ordered by: cumulative time
   List reduced from 108 to 10 due to restriction <10>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        2    0.000    0.000    0.547    0.273 /usr/lib/python3.12/asyncio/base_events.py:1922(_run_once)
        2    0.000    0.000    0.510    0.255 /usr/lib/python3.12/selectors.py:451(select)
        1    0.510    0.510    0.510    0.510 /tmp/ipython-input-2041063726.py:15(matmul_naive)
        2    0.000    0.000    0.037    0.019 /usr/lib/python3.12/asyncio/events.py:86(_run)
        2    0.000    0.000    0.037    0.019 {method 'run' of '_contextvars.Context' objects}
        1    0.000    0.000    0.037    0.037 /usr/local/lib/python3.12/dist-packages/tornado/platform/asyncio.py:206(_handle_events)
        1    0.000    0.000    0.037    0.037 /usr/local/lib/python3.12/dist-packages/zmq/eventloop/zmqstream.py:574(_handle_events)
        1    0.000    0.000    0.037    0.037 /usr/local/lib/python3.12/dist-packages/zmq/eventloop/zmqstream.py:615(_handle_recv)
        1    0.000    0.000    0.037    0.037 /usr/local/lib/python3.12/dist-packages/zmq/eventloop/zmqstream.py:547(_run_callback)
        1    0.000    0.000    0.037    0.037 /usr/local/lib/python3.12/dist-packages/ipykernel/iostream.py:107(_handle_event)



=== Memory Profiling ===
Current memory usage : 0.01 MB
Peak memory usage    : 1.29 MB
```

GoogleColab – TPU:



```
Naïve Matrix Multiplication Performance Analysis
•••
=== Execution Time ===
Matrix size : 200 x 200
Time taken  : 0.5277 seconds

=== CPU Profiling (Top 10) ===
         490 function calls (486 primitive calls) in 0.535 seconds

   Ordered by: cumulative time
   List reduced from 123 to 10 due to restriction <10>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        3    0.000    0.000    0.535    0.178 /usr/lib/python3.12/asyncio/base_events.py:1922(_run_once)
        3    0.000    0.000    0.519    0.173 /usr/lib/python3.12/selectors.py:451(select)
        1    0.313    0.313    0.313    0.313 /tmp/ipython-input-2041063726.py:15(matmul_naive)
        2    0.205    0.103    0.205    0.103 {method 'poll' of 'select.epoll' objects}
        3    0.000    0.000    0.016    0.005 /usr/lib/python3.12/asyncio/events.py:86(_run)
        3    0.000    0.000    0.016    0.005 {method 'run' of '_contextvars.Context' objects}
        2    0.000    0.000    0.016    0.008 /usr/local/lib/python3.12/dist-packages/tornado/platform/asyncio.py:206(_handle_events)
        2    0.000    0.000    0.016    0.008 /usr/local/lib/python3.12/dist-packages/zmq/eventloop/zmqstream.py:574(_handle_events)
        2    0.000    0.000    0.016    0.008 /usr/local/lib/python3.12/dist-packages/zmq/eventloop/zmqstream.py:615(_handle_recv)
        2    0.000    0.000    0.016    0.008 /usr/local/lib/python3.12/dist-packages/zmq/eventloop/zmqstream.py:547(_run_callback)



=== Memory Profiling ===
Current memory usage : 0.01 MB
Peak memory usage    : 1.29 MB
```

Local (Thonny) :



```
Naïve Matrix Multiplication Performance Analysis

=== Execution Time ===
Matrix size : 200 x 200
Time taken  : 1.2282 seconds

=== CPU Profiling (Top 10) ===
         4 function calls in 1.191 seconds

   Ordered by: cumulative time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    1.191    1.191    1.195    1.195 <string>:15(matmul_naive)
        1    0.000    0.000    0.004    0.004 <string>:17(<listcomp>)
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
        1    0.000    0.000    0.000    0.000 {built-in method builtins.len}



=== Memory Profiling ===
Current memory usage : 0.01 MB
Peak memory usage    : 1.29 MB
```

TABLE :

	CPU	GPU	TPU	LOCAL
TIME TAKEN	1.7083	1.3892	0.5277	1.2282
CPU PROFILING	1.526	0.547	0.535	1.191