**Review Questions**

**Use Profiling**

1. **Define the following metrics and perform the following operations**
2. **Write a Python program using Python Lists**
3. **Write a Python program and NumPy**

**Matrix A = [[ 3.7827 3.3454 3.2341] , [ 2.2122 3.5678 3.9087] ,**

**[1.1234 2.8934, 5.9087]].**

**Matrix B = [[ 3.1234 3.0987 3.1234] , [ 2.1111 3.2222 3.3333] ,**

**[1.0987 1.3456, 5.1234]].**

**Matrix C = [[ 3.1243 3.0989 3.1256 ] , [ 2.6721 3.6785 3.9017] ,**

**[1.1254 2.8956, 5.9187]].**

(a)

import cProfile

import pstats

import io

# Define matrices

A = [[3.7827, 3.3454, 3.2341],

[2.2122, 3.5678, 3.9087],

[1.1234, 2.8934, 5.9087]]

B = [[3.1234, 3.0987, 3.1234],

[2.1111, 3.2222, 3.3333],

[1.0987, 1.3456, 5.1234]]

# Function to add matrices

def add\_matrices(M1, M2):

return [[M1[i][j] + M2[i][j] for j in range(len(M1[0]))] for i in range(len(M1))]

# Function to multiply matrices

def multiply\_matrices(M1, M2):

return [[sum(M1[i][k] \* M2[k][j] for k in range(len(M2))) for j in range(len(M2[0]))] for i in range(len(M1))]

# Profiling the matrix addition

def profile\_addition():

result\_add = add\_matrices(A, B)

return result\_add

# Profiling the matrix multiplication

def profile\_multiplication():

result\_mul = multiply\_matrices(A, B)

return result\_mul

# Create a stream for profiling results

pr = cProfile.Profile()

pr.enable()

# Run the functions you want to profile

profile\_addition()

profile\_multiplication()

pr.disable()

# Print profiling results

s = io.StringIO()

ps = pstats.Stats(pr, stream=s).sort\_stats(pstats.SortKey.TIME)

ps.print\_stats()

print(s.getvalue())

Output :

128 function calls in 0.000 seconds

Ordered by: internal time

ncalls tottime percall cumtime percall filename:lineno(function)

3 0.000 0.000 0.000 0.000 {built-in method builtins.compile}

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:3490(run\_code)

3 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:20(<listcomp>)

9 0.000 0.000 0.000 0.000 {built-in method builtins.sum}

36 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:20(<genexpr>)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\codeop.py:117(\_\_call\_\_)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\traitlets\traitlets.py:689(\_\_get\_\_)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:104(\_\_init\_\_)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:15(add\_matrices)

3 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:16(<listcomp>)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:287(helper)

6 0.000 0.000 0.000 0.000 {built-in method builtins.next}

6 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\compilerop.py:180(extra\_flags)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:141(\_\_exit\_\_)

3 0.000 0.000 0.000 0.000 {built-in method builtins.exec}

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:1(<module>)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:19(multiply\_matrices)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:132(\_\_enter\_\_)

6 0.000 0.000 0.000 0.000 {built-in method builtins.getattr}

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:23(profile\_addition)

17 0.000 0.000 0.000 0.000 {built-in method builtins.len}

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\traitlets\traitlets.py:651(get)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:3442(compare)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\4119834923.py:28(profile\_multiplication)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:1231(user\_global\_ns)

1 0.000 0.000 0.000 0.000 {method 'disable' of '\_lsprof.Profiler' objects}

(b)

import numpy as np

import cProfile

import pstats

import io

# Define matrices using NumPy arrays

A = np.array([[3.7827, 3.3454, 3.2341],

[2.2122, 3.5678, 3.9087],

[1.1234, 2.8934, 5.9087]])

B = np.array([[3.1234, 3.0987, 3.1234],

[2.1111, 3.2222, 3.3333],

[1.0987, 1.3456, 5.1234]])

# Function to add matrices

def add\_matrices\_np(M1, M2):

return M1 + M2

# Function to multiply matrices

def multiply\_matrices\_np(M1, M2):

return np.dot(M1, M2)

# Profiling the matrix addition

def profile\_addition\_np():

result\_add = add\_matrices\_np(A, B)

return result\_add

# Profiling the matrix multiplication

def profile\_multiplication\_np():

result\_mul = multiply\_matrices\_np(A, B)

return result\_mul

# Create a stream for profiling results

pr = cProfile.Profile()

pr.enable()

# Run the functions you want to profile

profile\_addition\_np()

profile\_multiplication\_np()

pr.disable()

# Print profiling results

s = io.StringIO()

ps = pstats.Stats(pr, stream=s).sort\_stats(pstats.SortKey.TIME)

ps.print\_stats()

print(s.getvalue())

Output :

63 function calls in 0.037 seconds

Ordered by: internal time

ncalls tottime percall cumtime percall filename:lineno(function)

1 0.037 0.037 0.037 0.037 {built-in method numpy.core.\_multiarray\_umath.implement\_array\_function}

3 0.000 0.000 0.000 0.000 {built-in method builtins.compile}

3 0.000 0.000 0.037 0.012 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:3490(run\_code)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3089373465.py:16(add\_matrices\_np)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\codeop.py:117(\_\_call\_\_)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:104(\_\_init\_\_)

3 0.000 0.000 0.037 0.012 {built-in method builtins.exec}

1 0.000 0.000 0.037 0.037 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3089373465.py:20(multiply\_matrices\_np)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3089373465.py:1(<module>)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:141(\_\_exit\_\_)

1 0.000 0.000 0.037 0.037 <\_\_array\_function\_\_ internals>:177(dot)

6 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\compilerop.py:180(extra\_flags)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:3442(compare)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:287(helper)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\traitlets\traitlets.py:689(\_\_get\_\_)

6 0.000 0.000 0.000 0.000 {built-in method builtins.getattr}

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\contextlib.py:132(\_\_enter\_\_)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\traitlets\traitlets.py:651(get)

6 0.000 0.000 0.000 0.000 {built-in method builtins.next}

1 0.000 0.000 0.037 0.037 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3089373465.py:29(profile\_multiplication\_np)

3 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:1231(user\_global\_ns)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3089373465.py:24(profile\_addition\_np)

1 0.000 0.000 0.000 0.000 {method 'disable' of '\_lsprof.Profiler' objects}

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\numpy\core\multiarray.py:740(dot)

1. **Write a Python Program and perform the following operations ?**
2. Create a List {1225, 4986, 6789, 7890, 2345, 6783, 0987, 1234, 8765, 3456}
3. Iterate using a for loop
4. Iterate using for loop and range
5. List Comprehension
6. Enumerate
7. Iter function and next function
8. Map function
9. Using zip
10. Using NumPy Module

import cProfile

import pstats

import io

import numpy as np

# Define the list

data = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]

# I. Create a List

def create\_list():

return [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]

# II. Iterate using a for loop

def iterate\_for\_loop(lst):

for item in lst:

pass # Replace pass with any operation if needed

# III. Iterate using for loop and range

def iterate\_for\_loop\_range(lst):

for i in range(len(lst)):

item = lst[i]

pass # Replace pass with any operation if needed

# IV. List Comprehension

def list\_comprehension(lst):

return [x \*\* 2 for x in lst]

# V. Enumerate

def enumerate\_function(lst):

for index, item in enumerate(lst):

pass # Replace pass with any operation if needed

# VI. Iter function and next function

def iter\_and\_next(lst):

iterator = iter(lst)

for \_ in lst:

item = next(iterator)

# VII. Map function

def map\_function(lst):

return list(map(lambda x: x \*\* 2, lst))

# VIII. Using zip (Pairing the list with itself for demonstration)

def zip\_function(lst):

return list(zip(lst, lst))

# IX. Using NumPy Module

def numpy\_operations(lst):

np\_data = np.array(lst)

return np\_data \*\* 2

# Profiling all operations

def profile\_operations():

pr = cProfile.Profile()

pr.enable()

create\_list()

iterate\_for\_loop(data)

iterate\_for\_loop\_range(data)

list\_comprehension(data)

enumerate\_function(data)

iter\_and\_next(data)

map\_function(data)

zip\_function(data)

numpy\_operations(data)

pr.disable()

# Print profiling results

s = io.StringIO()

ps = pstats.Stats(pr, stream=s).sort\_stats(pstats.SortKey.TIME)

ps.print\_stats()

print(s.getvalue())

# Run the profiling

profile\_operations()

Output:

34 function calls in 0.000 seconds

Ordered by: internal time

ncalls tottime percall cumtime percall filename:lineno(function)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:48(numpy\_operations)

1 0.000 0.000 0.000 0.000 {built-in method numpy.array}

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:29(enumerate\_function)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:19(iterate\_for\_loop\_range)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:34(iter\_and\_next)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:40(map\_function)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:26(<listcomp>)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:44(zip\_function)

10 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:41(<lambda>)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:25(list\_comprehension)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:14(iterate\_for\_loop)

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1754337465.py:10(create\_list)

10 0.000 0.000 0.000 0.000 {built-in method builtins.next}

1 0.000 0.000 0.000 0.000 {built-in method builtins.len}

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.iter}

1. **For a List A, B, C, D, E write a python program to compute all the combinations and permutations**

import cProfile

import pstats

import io

import itertools

# Define lists

A = [1, 2]

B = [3, 4]

C = [5, 6]

D = [7, 8]

E = [9, 10]

# Function to compute permutations

def compute\_permutations(lst):

perms = list(itertools.permutations(lst))

print(f"Permutations of {lst}:")

print(perms)

return perms

# Function to compute combinations

def compute\_combinations(lst, r):

combs = list(itertools.combinations(lst, r))

print(f"Combinations of {lst} with length {r}:")

print(combs)

return combs

# Profiling permutations

def profile\_permutations():

print("\n--- Permutations ---")

perms\_A = compute\_permutations(A)

perms\_B = compute\_permutations(B)

perms\_C = compute\_permutations(C)

perms\_D = compute\_permutations(D)

perms\_E = compute\_permutations(E)

# Profiling combinations

def profile\_combinations():

print("\n--- Combinations ---")

combs\_A2 = compute\_combinations(A, 2)

combs\_B2 = compute\_combinations(B, 2)

combs\_C2 = compute\_combinations(C, 2)

combs\_D2 = compute\_combinations(D, 2)

combs\_E2 = compute\_combinations(E, 2)

# Profiling all operations

def profile\_operations():

pr = cProfile.Profile()

pr.enable()

profile\_permutations()

profile\_combinations()

pr.disable()

# Print profiling results

s = io.StringIO()

ps = pstats.Stats(pr, stream=s).sort\_stats(pstats.SortKey.TIME)

ps.print\_stats()

print("\n--- Profiling Results ---")

print(s.getvalue())

# Run the profiling

profile\_operations()

Output:

--- Permutations ---

Permutations of [1, 2]:

[(1, 2), (2, 1)]

Permutations of [3, 4]:

[(3, 4), (4, 3)]

Permutations of [5, 6]:

[(5, 6), (6, 5)]

Permutations of [7, 8]:

[(7, 8), (8, 7)]

Permutations of [9, 10]:

[(9, 10), (10, 9)]

--- Combinations ---

Combinations of [1, 2] with length 2:

[(1, 2)]

Combinations of [3, 4] with length 2:

[(3, 4)]

Combinations of [5, 6] with length 2:

[(5, 6)]

Combinations of [7, 8] with length 2:

[(7, 8)]

Combinations of [9, 10] with length 2:

[(9, 10)]

--- Profiling Results ---

395 function calls in 0.002 seconds

Ordered by: internal time

ncalls tottime percall cumtime percall filename:lineno(function)

1 0.001 0.001 0.001 0.001 C:\Users\manid\anaconda3\Lib\threading.py:1118(\_wait\_for\_tstate\_lock)

44 0.001 0.000 0.002 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:610(write)

22 0.000 0.000 0.002 0.000 {built-in method builtins.print}

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\zmq\sugar\socket.py:543(send)

1 0.000 0.000 0.000 0.000 {method 'acquire' of '\_thread.lock' objects}

1 0.000 0.000 0.001 0.001 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:243(schedule)

44 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:505(\_is\_master\_process)

5 0.000 0.000 0.001 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3050118737.py:21(compute\_combinations)

5 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3050118737.py:14(compute\_permutations)

44 0.000 0.000 0.001 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:532(\_schedule\_flush)

44 0.000 0.000 0.000 0.000 {method 'write' of '\_io.StringIO' objects}

1 0.000 0.000 0.001 0.001 C:\Users\manid\anaconda3\Lib\threading.py:1185(is\_alive)

44 0.000 0.000 0.000 0.000 {method '\_\_exit\_\_' of '\_thread.RLock' objects}

44 0.000 0.000 0.000 0.000 {built-in method nt.getpid}

1 0.000 0.000 0.002 0.002 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3050118737.py:28(profile\_permutations)

44 0.000 0.000 0.000 0.000 {built-in method builtins.len}

44 0.000 0.000 0.000 0.000 {built-in method builtins.isinstance}

1 0.000 0.000 0.001 0.001 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\3050118737.py:37(profile\_combinations)

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:127(\_event\_pipe)

1 0.000 0.000 0.000 0.000 {method 'disable' of '\_lsprof.Profiler' objects}

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\threading.py:568(is\_set)

1 0.000 0.000 0.000 0.000 {method 'append' of 'collections.deque' objects}

1. **Using the same list use itertools compute permutations and combinations**

import cProfile

import pstats

import io

import itertools

# Define lists

A = [1, 2]

B = [3, 4]

C = [5, 6]

D = [7, 8]

E = [9, 10]

# Function to compute permutations

def compute\_permutations(lst):

perms = list(itertools.permutations(lst))

print(f"Permutations of {lst}:")

for perm in perms:

print(perm)

return perms

# Function to compute combinations

def compute\_combinations(lst, r):

combs = list(itertools.combinations(lst, r))

print(f"Combinations of {lst} with length {r}:")

for comb in combs:

print(comb)

return combs

# Profiling permutations

def profile\_permutations():

print("\n--- Permutations ---")

compute\_permutations(A)

compute\_permutations(B)

compute\_permutations(C)

compute\_permutations(D)

compute\_permutations(E)

# Profiling combinations

def profile\_combinations():

print("\n--- Combinations ---")

compute\_combinations(A, 2)

compute\_combinations(B, 2)

compute\_combinations(C, 2)

compute\_combinations(D, 2)

compute\_combinations(E, 2)

# Profiling all operations

def profile\_operations():

pr = cProfile.Profile()

pr.enable()

profile\_permutations()

profile\_combinations()

pr.disable()

# Print profiling results

s = io.StringIO()

ps = pstats.Stats(pr, stream=s).sort\_stats(pstats.SortKey.TIME)

ps.print\_stats()

print("\n--- Profiling Results ---")

print(s.getvalue())

# Run the profiling

profile\_operations()

Output:

--- Permutations ---

Permutations of [1, 2]:

(1, 2)

(2, 1)

Permutations of [3, 4]:

(3, 4)

(4, 3)

Permutations of [5, 6]:

(5, 6)

(6, 5)

Permutations of [7, 8]:

(7, 8)

(8, 7)

Permutations of [9, 10]:

(9, 10)

(10, 9)

--- Combinations ---

Combinations of [1, 2] with length 2:

(1, 2)

Combinations of [3, 4] with length 2:

(3, 4)

Combinations of [5, 6] with length 2:

(5, 6)

Combinations of [7, 8] with length 2:

(7, 8)

Combinations of [9, 10] with length 2:

(9, 10)

--- Profiling Results ---

480 function calls in 0.001 seconds

Ordered by: internal time

ncalls tottime percall cumtime percall filename:lineno(function)

54 0.000 0.000 0.001 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:610(write)

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\zmq\sugar\socket.py:543(send)

27 0.000 0.000 0.001 0.000 {built-in method builtins.print}

54 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:505(\_is\_master\_process)

5 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1215979415.py:14(compute\_permutations)

54 0.000 0.000 0.000 0.000 {built-in method nt.getpid}

5 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1215979415.py:22(compute\_combinations)

54 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:532(\_schedule\_flush)

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:243(schedule)

54 0.000 0.000 0.000 0.000 {method '\_\_exit\_\_' of '\_thread.RLock' objects}

54 0.000 0.000 0.000 0.000 {method 'write' of '\_io.StringIO' objects}

1 0.000 0.000 0.001 0.001 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1215979415.py:30(profile\_permutations)

54 0.000 0.000 0.000 0.000 {built-in method builtins.len}

54 0.000 0.000 0.000 0.000 {built-in method builtins.isinstance}

1 0.000 0.000 0.000 0.000 C:\Users\manid\AppData\Local\Temp\ipykernel\_11672\1215979415.py:39(profile\_combinations)

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\threading.py:1185(is\_alive)

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\threading.py:1118(\_wait\_for\_tstate\_lock)

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\site-packages\ipykernel\iostream.py:127(\_event\_pipe)

1 0.000 0.000 0.000 0.000 {method 'acquire' of '\_thread.lock' objects}

1 0.000 0.000 0.000 0.000 C:\Users\manid\anaconda3\Lib\threading.py:568(is\_set)

1 0.000 0.000 0.000 0.000 {method 'disable' of '\_lsprof.Profiler' objects}

1 0.000 0.000 0.000 0.000 {method 'append' of 'collections.deque' objects}