

Problem 5 - Python

Problem 5a: Dot Product of Two Vectors

```
# Dot Product of 2 vectors

from time import time

import numpy as np

n = 10000 # Length of the vectors

a = np.random.rand(n, 1) # Create a random column vector a
b = np.random.rand(n, 1) # Create a random column vector b

c = 0 # pre-allocate memory to store the result in c

# Dot product with for-loop
t1 = time()
for i in range(n):
    c += a[i] * b[i]
t2 = time()

time_loop = t2 - t1

# Dot product with numpy
t3 = time()

cc = np.vdot(a, b)

t4 = time()

time_numpy = t4 - t3

# Compare the results
norm = np.linalg.norm(cc - c)

# Measure the speed-up
speedup = time_loop / time_numpy

print("n: ", n)
print("norm: ", norm)
print("Speedup: ", speedup)
```

```
(.venv) → Assignment_1 python3 p5_vv_prod.py
n: 10000
norm: 7.73070496506989e-12
Speedup: 713.2181818181818
```

Problem 5b: Matrix-Vector Product

```
from time import time

import numpy as np

n = 10000

A = np.random.rand(n, n) # Create a Matrix of size n,n
b = np.random.rand(n, 1) # Create a random column vector b

c = np.zeros((n, 1)) # pre-allocate memory to store the result in c

# Dot product with numpy
t1 = time()

cc = np.dot(A, b)

t2 = time()

time_numpy = t2 - t1

# Dot product with for-loop
t3 = time()
for i in range(n):
    for j in range(n):
        c[i] += A[i][j] * b[j]

t4 = time()

time_loop = t4 - t3

# Compare the results
norm = np.linalg.norm(cc - c)

# Measure the speed-up
speedup = time_loop / time_numpy

print("n: ", n)
print("norm: ", norm)
print("Speedup: ", speedup)
```

```
(.venv) → Assignment_1 python3 p5_mv_prod.py
n: 10000
norm: 7.231167827724513e-10
Speedup: 835.9181061974602
```

Problem 5c: Matrix-matrix Product

```
from time import time

import numpy as np

n = 1000

A = np.random.rand(n, n) # Create a Matrix A of size n,n
B = np.random.rand(n, n) # Create a Matrix B of size n,n

c = np.zeros((n, n)) # pre-allocate memory to store the result in c

# Dot product with numpy
t1 = time()

cc = np.dot(A, B)

t2 = time()

time_numpy = t2 - t1

# Dot product with for-loop
t3 = time()
for i in range(n):
    for j in range(n):
        for k in range(n):
            c[i][j] += A[i][k] * B[k][j]

t4 = time()

time_loop = t4 - t3

# Compare the results
norm = np.linalg.norm(cc - c)

# Measure the speed-up
speedup = time_loop / time_numpy

print("n: ", n)
print("norm: ", norm)
print("Speedup: ", speedup)
```

```
(.venv) → Assignment_1 python3 p5_mm_prod.py
n: 1000
norm: 2.0366650726596068e-10
Speedup: 35765.234444432885
```

