ES215 - COA - Assignment-4

Name: Daniel Giftson E

Roll No: 20110051

Q4)

- From Bucket 2, I coded Matrix Multiplication in Python. (Github link to the code)
- I have specified N as an input variable (n).
- I have made minor changes to the code I submitted for Assignment-1 to easily work on this question. I have shared the modified code in the zip file.
- Instead of directly using **np.dot()** to find the resultant matrix, I coded with three loops according to the question.
- I have also initialized the matrix elements to be 0 by using **np.zeros()**.
- I have compiled and run this Python code in Google Colab.

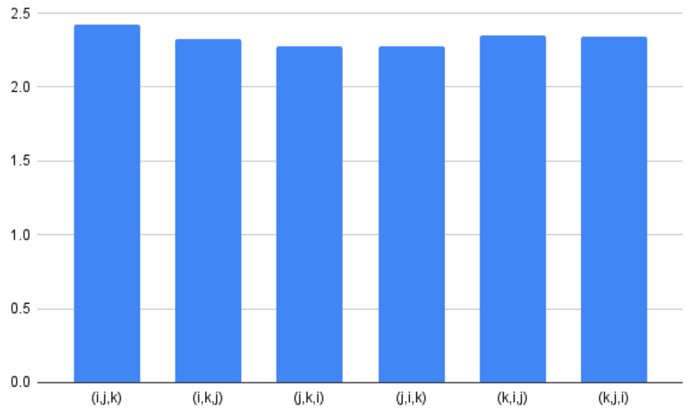
a)

- I have represented the 6 combinations of i,j,k as a tuple (i,j,k) where,
  i = outer loop variable
  j = middle loop variable
  k = inner loop variable

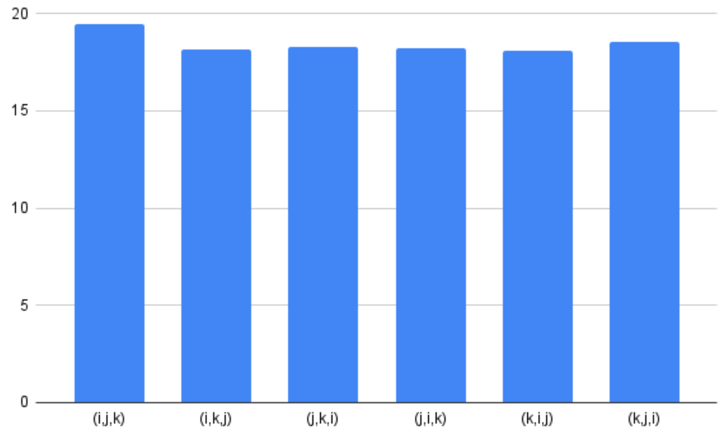
Loop Combination	N = 128	N = 256	N = 512
(i,j,k)	2.420335 secs	19.419688 secs	148.215148 secs
(i,k,j)	2.325594 secs	18.117965 secs	146.717773 secs
(j,k,i)	2.276167 secs	18.289763 secs	150.798155 secs
(j,i,k)	2.278422 secs	18.214583 secs	146.252822 secs
(k,i,j)	2.348039 secs	18.067528 secs	147.942867 secs
(k,j,i)	2.343067 secs	18.530496 secs	150.940850 secs

b)

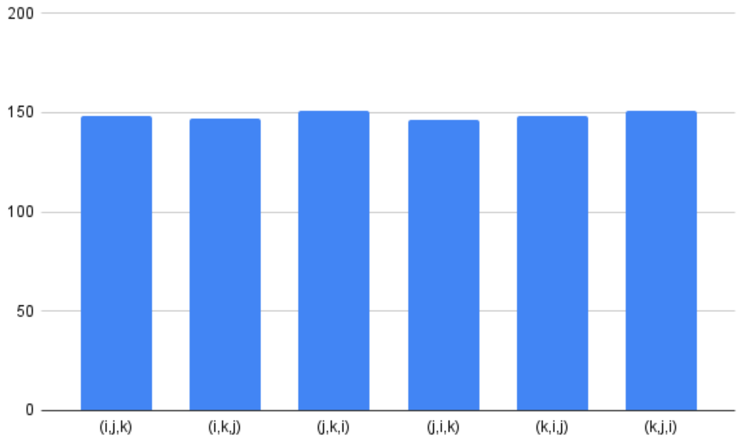
- **Plot of execution time for each iteration: (N = 128)**



- **Plot of execution time for each iteration: (N = 256)**



- **Plot of execution time for each iteration: (N = 512)**



Observations:

- We observe that the execution time increases for each iteration as the value of N increases.
- We also observe that the execution time is very much higher and more significant in the case of Python compared to C++.
- We also observe that for a given value of N, the execution time of each iteration is almost equal to each other compared to a changing execution time in the case of C++.

Inference:

- We conclude that as we interchange the position of the three loops, the performance/execution time still/almost remains the same in the case of Python.