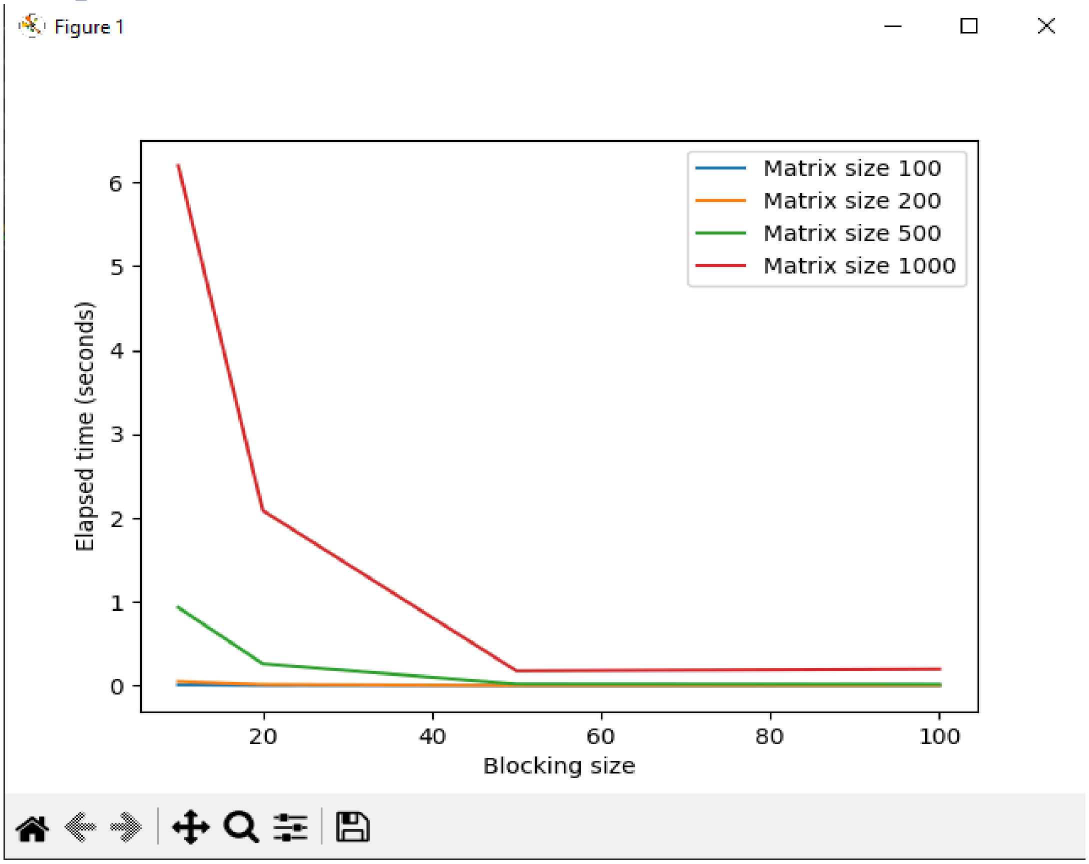
Matrix multiplication by blocking

Name:AbdElRahman Tarek
Id:7427 g1

Code

You can check the code within the zip.

Output of code




```
Run:  main
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\main.py
Matrix size: 100 | Block size: 10 | Elapsed time: 0.0090 seconds
Matrix size: 100 | Block size: 20 | Elapsed time: 0.0015 seconds
Matrix size: 100 | Block size: 50 | Elapsed time: 0.0000 seconds
Matrix size: 100 | Block size: 100 | Elapsed time: 0.0000 seconds
Matrix size: 200 | Block size: 10 | Elapsed time: 0.0509 seconds
Matrix size: 200 | Block size: 20 | Elapsed time: 0.0140 seconds
Matrix size: 200 | Block size: 50 | Elapsed time: 0.0010 seconds
Matrix size: 200 | Block size: 100 | Elapsed time: 0.0020 seconds
Matrix size: 500 | Block size: 10 | Elapsed time: 0.9329 seconds
Matrix size: 500 | Block size: 20 | Elapsed time: 0.2610 seconds
Matrix size: 500 | Block size: 50 | Elapsed time: 0.0199 seconds
Matrix size: 500 | Block size: 100 | Elapsed time: 0.0169 seconds
Matrix size: 1000 | Block size: 10 | Elapsed time: 6.1995 seconds
Matrix size: 1000 | Block size: 20 | Elapsed time: 2.0874 seconds
Matrix size: 1000 | Block size: 50 | Elapsed time: 0.1757 seconds
Matrix size: 1000 | Block size: 100 | Elapsed time: 0.1973 seconds
```

Discussion

Each line in the plot corresponds to a different matrix size, indicating which matrix size each line represents. The x-axis represents the blocking size used for matrix multiplication, and the y-axis represents the elapsed time in seconds.

As we can see from the plot the elapsed time generally decreases as the blocking size increases that is due to large blocking size result few cache misses that's why we use blocking in matrix multiplication because if we use the common way it will cause cache misses and each time access the memory [slow speed].

From the plot we can say that array sizes **100** and **200** are much faster with optimal block size between **20-50** as we increase larger block size it will cause slower performance.

Larger array sizes **500-1000** are much faster with optimal block size between **50-100** same as before as we increase larger block size it will cause slower performance.

We conclude the most optimal blocking size with least cache misses depends on the matrix size, the benefits of blocking are most important for larger matrix sizes where the cache misses is larger.