EE451 PHW1

Jiajin Zhang 8459548972
Windows 11, VScode, mingw64_8.1.0

P1a.

Running on Windows 2 times, and here are the results.

```
PS C:\USC\24Fall\EE451> cd "c:\USC\24Fall\EE451\PHM1\" ; if ($?) { g++ p1a.cpp -o p1a } ; if ($?) { .\p1a
}
Number of FLOPs = 0, Execution time = 634.838163 sec,
216.494473 MFLOPs per sec
C[100][100]=879616000.000000
```

```
PS C:\USC\24Fall\EE451\PHM1> cd "c:\USC\24Fall\EE451\PHM1\" ; if ($?) { g++ p1a.cpp -o p1a } ; if ($?) { .
\p1a }
Number of FLOPs = 0, Execution time = 552.012729 sec,
248.977870 MFLOPs per sec
C[100][100]=879616000.000000
```

Running #	Execution time
1	634.838163 sec
2	552.012729 sec
Average	593.425446 sec

P1b.

Running on Windows.

For block size = 4,

```
PS C:\USC\24Fall\EE451\PHM1> cd "c:\USC\24Fall\EE451\PHM1\" ; if ($?) { g++ p1b.cpp -o p1b } ; if ($?) { .
\p1b }
block size = 4Number of FLOPs = 0, Execution time = 299.799321 sec,
458.436507 MFLOPs per sec
C[100][100]=879616000.000000
```

For block size = 8,

```
PS C:\USC\24Fall\EE451\PHM1> cd "c:\USC\24Fall\EE451\PHM1\" ; if ($?) { g++ p1b.cpp -o p1b } ; if ($?) { .
\p1b }
block size = 8Number of FLOPs = 0, Execution time = 244.408178 sec,
562.333693 MFLOPs per sec
C[100][100]=879616000.000000
```

For block size = 16,

```
PS C:\USC\24Fall\EE451\PHM1> cd "c:\USC\24Fall\EE451\PHM1\" ; if ($?) { g++ p1b.cpp -o p1b } ; if ($?) { .
\p1b }
block size = 16Number of FLOPs = 0, Execution time = 204.753315 sec,
671.241654 MFLOPs per sec
C[100][100]=879616000.000000
```

Number of blocks	4	8	16
Execution time	299.799321 sec	244.408178 sec	204.753315 sec

Based on the test results presented above, we can see that with the increase of block size, the code is running faster. Comparing with the naïve matrix multiplication, no matter how many the block size is, the execution time decreases in a significant way. In general, we can say that the block matrix multiplication is more efficient than naïve matrix multiplication.

P2.

Running on windows and the output is:

```
C:\USC\24Fall\EE451\PHM1>gcc -fopenmp -o p2 p2.cpp -lm

C:\USC\24Fall\EE451\PHM1>p2.exe
Execution time: 0.0290 seconds
```

And this is the output image: