Q1)

For Strassen.cpp, I modify the code for larger size 256,512,1024 as the original code don’t have enough memory to do the operation.

Dynamic memory allocation:

A computer screen shot of text

Description automatically generated

A screen shot of a computer

Description automatically generated

The full source code will be attached.

Performance comparison:

256:

|  |  |
| --- | --- |
| A computer screen shot of a computer  Description automatically generated | A screenshot of a computer  Description automatically generated |

512

|  |  |
| --- | --- |
|  |  |

1024

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated |  |

As you can see, the result is not good on Strassen algorithm. The main result is the matrix multiplication is now using dynamic memory allocation. When we create a sub matrix like 2x2, it will call new and those is bad on latency and for bigger the size like 512 and 1024, it is really bad for the memory access, so it take a lot of time.

Q2) im2cA screen shot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

Q3

Winograd:  
A screenshot of a computer program

Description automatically generated

Result: im2c is faster than winograd

A screenshot of a computer

Description automatically generatedA computer screen shot of a computer

Description automatically generated

Reason:

Winograd's algorithm reduces the number of multiplications at the expense of increasing the number of additions and memory accesses. Winograd algorithm is typically more beneficial for larger convolution kernels. For small kernels (like 3x3), the performance gain may not be significant or may even be negative compared to im2col.so im2col is outperform in this setting.