Matrix matrix multiplication:

Suppose there are two matrices A, B with dimentions as m x n and n x p respectively.
Now we know that the result will be another matrix C with dimention m x p.

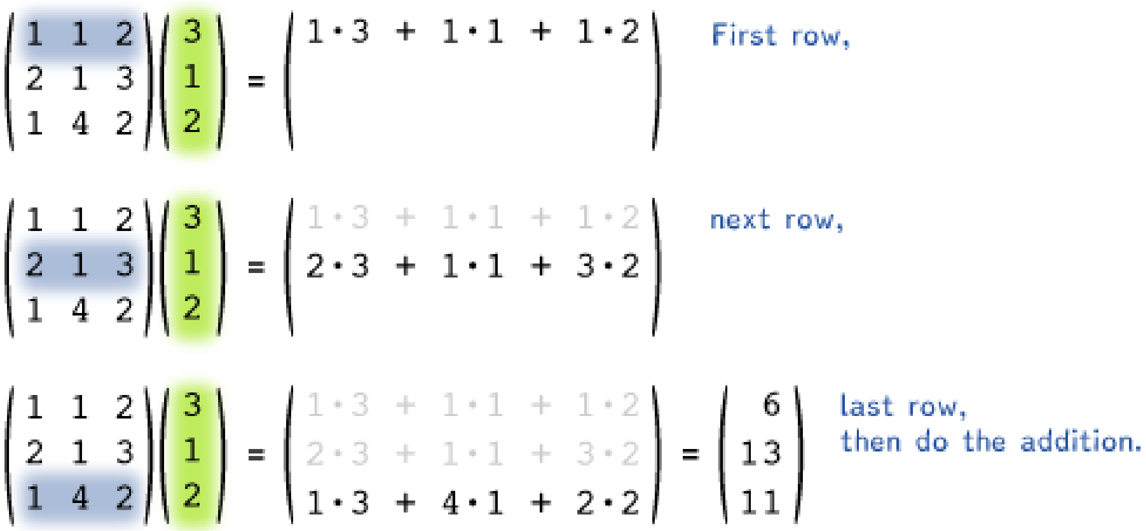
For the sequential method the complexity of the method to compute the multiplication will be $O(n^3)$ that is order of mnp.

Method:

```
loop i from 0 to m-1
        loop j from 0 to p – 1
                loop k from 0 to n-1
                        compute C[i][j] += A[i][k]*B[k][j];
                end loop
        end loop
end loop
```

parallelization:

To parallelize the multiplication of two matrices we need to apply the concept of matrix vector multiplication. In matrix vector multiplication the A matrix can be visualized as a collection of m vectors of size n and in the end matrix vector multiplication is reduced to simultaneous m dot product between the vectors.



So, Now in matrix matrix multiplicatin we can thing of the B matrix as a collection of column vectors. And similarly with the matrix vector multiplication we are basicaly calculating the dot product of different vectors to generate the result.

So row vectors of the first matrix and the column vectors of the second matrix is responsible for the result of the matrix matrix multiplication in the following way:

- The i, j th position of the result matrix that is C. Is nothing but the dot product of ith row vector of the A matrix and j th column vector of B matrix.

So, one of the methology will be

- At first transpose the B matrix creating a collection of row vectors for the ease of access, creating a p x n matrix as B had dimention of n x p .
- Now pass part of the both matrix to different processes. Say there are P processes, so pass m/p number of rows from the A matrix and p/P number of rows from the B matrix to each process.
- Compute the local dot product of row vectors of each submatrix. The dot product between i th row vector of local submatrix of A and j th row of local submatrix of B create the result of the (i,j)th position of C matrix.