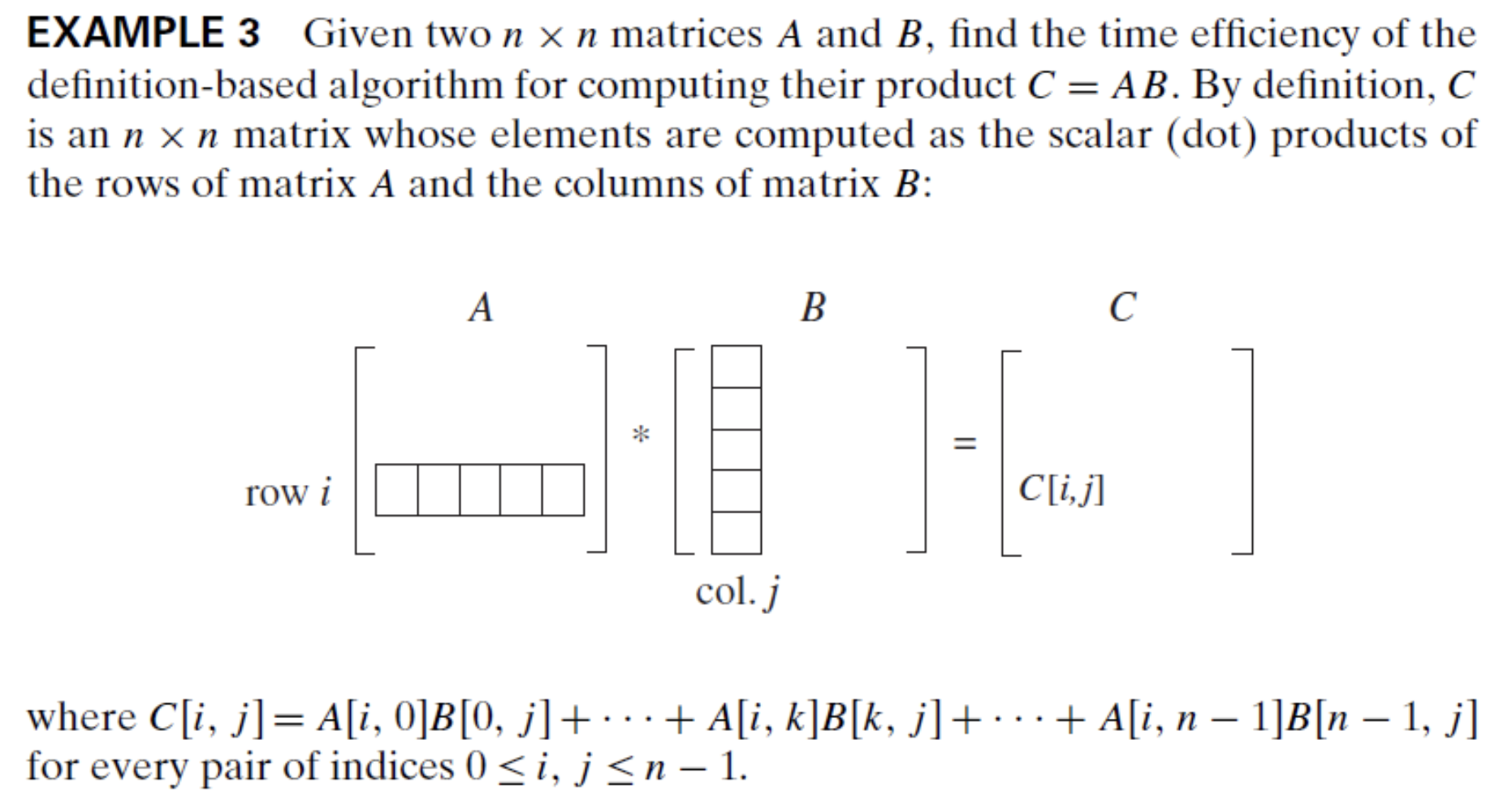
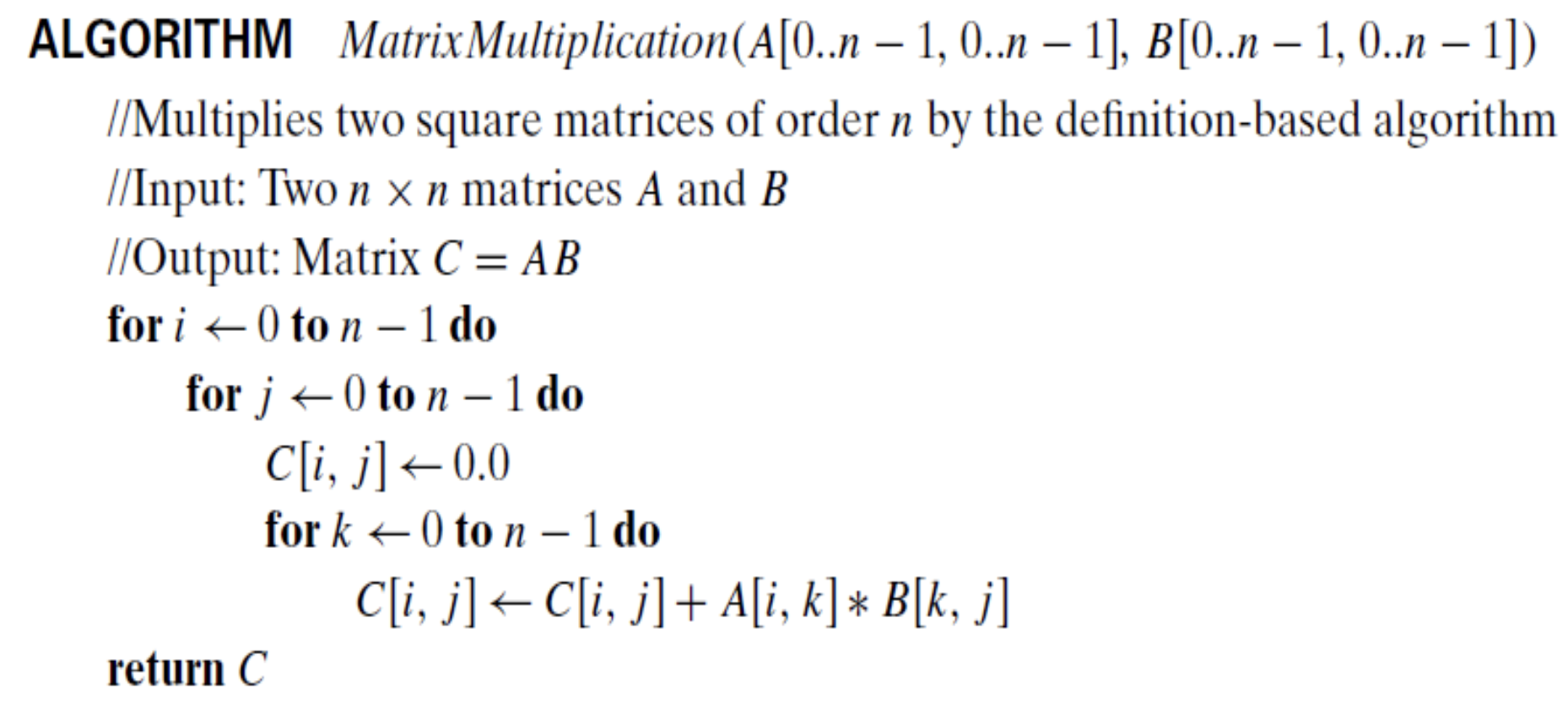
Efficiency Analysis of Non-Recursive Algorithms

Additional exercise

# Problem



# Algorithm



Using the General Plan for Efficiency Analysis, determine the order of growth of that algorithm.

You can find the solution in the next page.

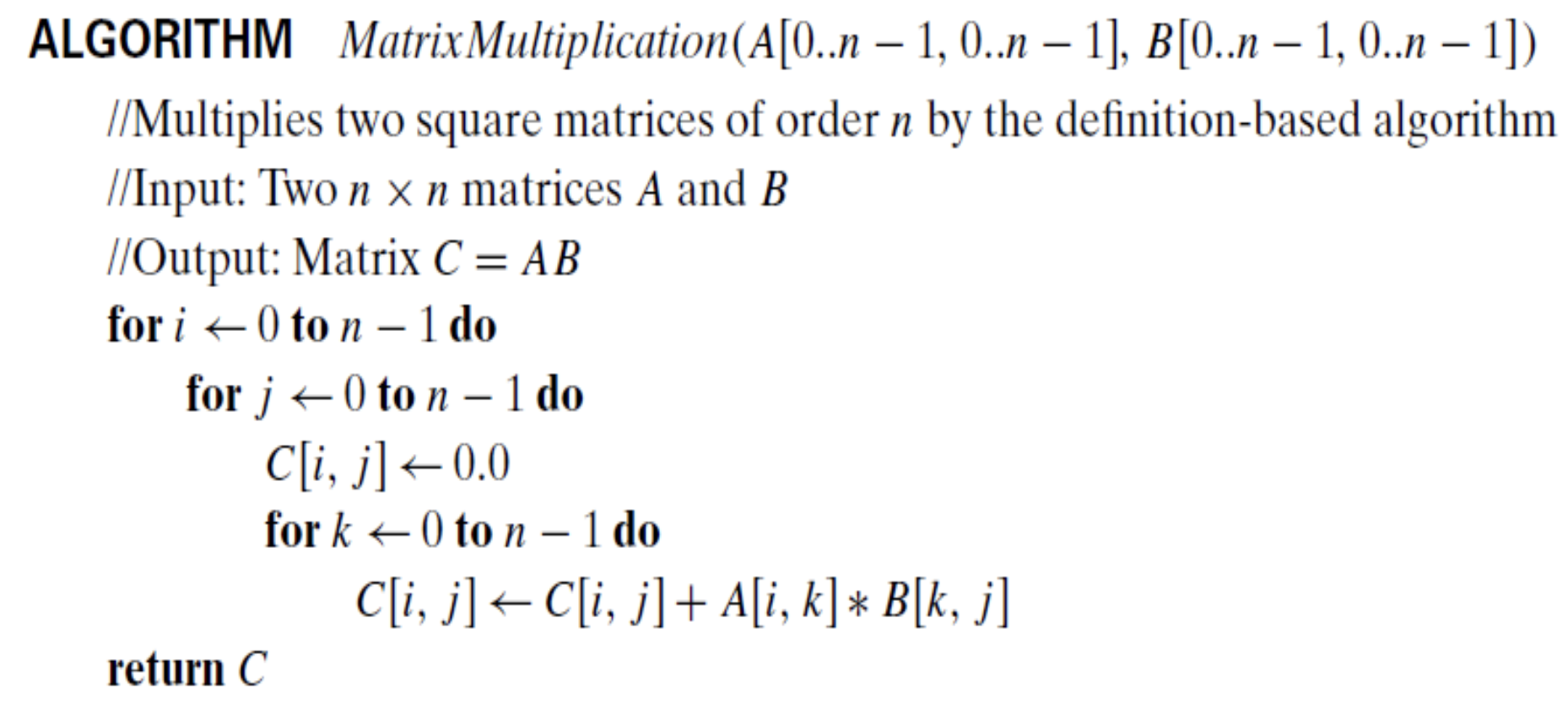
# General Plan for Efficiency Analysis

## Decide on a parameter indicating an input’s size

From the input matrix A (array), use its size i.e. n which represents the number of rows (or columns) in one matrix.

## Identify the algorithm’s basic operation

The algorithm’s inner loop is



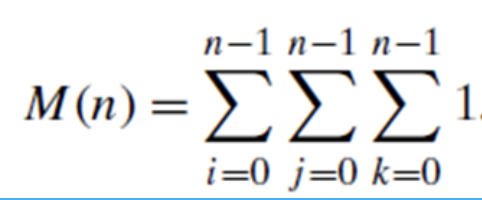
It contains 3 operations: an assignment, a multiplication, and a sum. We’ll use the multiplication as the basic operation. We could also use the others (sum or assignment) since they repeat the same number of times.

## Check whether the number of times the basic operation is executed depends also on some additional property. If so, worst-case, average-case, and best-case efficiencies have to be investigated

The number of assignments will be the same for all matrices of size n x n. Therefore, in terms in this metric, there is no need to distinguish among the worst, average, and best cases

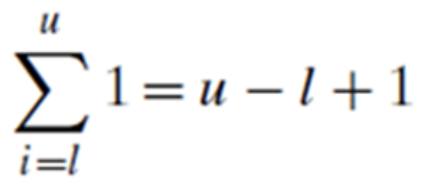
## Set up a sum expressing the number of times the algorithm’s basic operation is executed

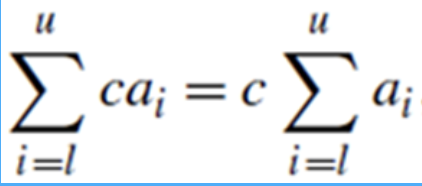
Each loop becomes a summation in sigma notation.



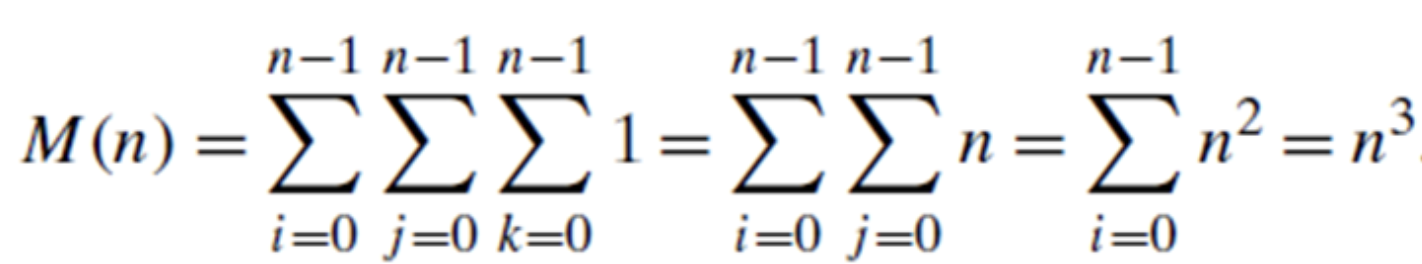
## Find a closed-form formula for the count or, at the very least, establish its order of growth

Using these formulas:





We solve each sigma notation from the inner one to the outer one.



Now we can classify its order of growth.