1 Parallel Summation

For the parallel summation of $N^2$ integer numbers a 2-dimensional torus of size $N \times N$ is used, where each computing node stores exactly one integer $val \leftarrow rank + 1$. The parallel summation algorithm works as follows:

```plaintext
code
sum \leftarrow val
tmp \leftarrow val

for i \leftarrow 1 to N-1 do
    MPI_send(tmp, 1, MPI_INT, left neighbour, ...)
    MPI_recv(tmp, 1, MPI_INT, right neighbour, ...)
    sum \leftarrow sum + tmp
od

tmp \leftarrow sum

for i \leftarrow 1 to N-1 do
    MPI_send(tmp, 1, MPI_INT, lower neighbour, ...)
    MPI_recv(tmp, 1, MPI_INT, higher neighbour, ...)
    sum \leftarrow sum + tmp
od
```

a) Describe how this algorithm works for a 3×3 torus, where the nodes are labelled rowise (starting with rank 0) from top left to bottom right.

![3x3 torus diagram]

b) Dependent on $N$ only, determine the number of necessary computing steps (i.e. $T(p)$) of the parallel program (communication is to be neglected) and, thus, give an approximation for the speed-up and parallel efficiency. Sketch the parallel efficiency in a small diagram and discuss these results!

c) Implement the above program using MPI and test it for different values of $N$, running your code on the cluster of the Chair for Computation in Engineering.