DSAI 3202 – Parallel and distributed computing

Assignment 2 – Part 1/3: Game of Life

This exercise is worth 40 points on your assignment.

# Objectives:

* Develop Python programs that run the game of life in a distributed fashion using MPI4PY.

# Tools and Concepts:

* Python: Programming language.
* Packages: MPI4PY.

# The game of life

## Description

The Game of Life, devised by mathematician John Conway, is a cellular automaton simulation. It consists of a grid of cells, each of which can be in one of two states: alive (■) or dead (□). The state of each cell at the next time step is determined by the following rules:

1. Any live cell with fewer than two live neighbors dies, as if by underpopulation.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

The Game of Life is an example of a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input from human players.

## Use of the game of life

The Game of Life, despite its simplicity, serves as a fascinating example of cellular automata and has several purposes and uses in various fields:

### Mathematical Exploration:

The Game of Life is a rich subject for mathematical analysis and has led to the discovery of various patterns and structures, such as still lifes, oscillators, and spaceships. It provides a playground for exploring concepts in geometry, algebra, and theoretical biology.

### Biological and Physical Modeling:

While the Game of Life is not a perfect model for any real biological or physical system, it is an example of how simple rules can lead to complex behavior. It is used as a metaphor for processes such as the growth of bacterial colonies, spread of diseases, and formation of biological patterns.

### Philosophical Inquiry:

The Game of Life raises philosophical questions about the nature of life and the universe. It shows how life-like behavior can emerge from simple, non-living components, contributing to discussions about the definition of life and the possibility of artificial life.

# Tasks

### Initialize MPI:

* Import the MPI4PY module and initialize MPI.
* Determine the rank and size of the MPI processes.

### Define the Grid:

* Set the dimensions of the grid (e.g., 20x20).
* Divide the grid into horizontal strips, assigning one strip to each MPI process.

### Initialize the Grid:

* Randomly initialize the local grid (the strip assigned to each process) with live and dead cells.

### Define the Update Function:

* Implement a function to update the local grid based on the Game of Life rules. This function should take into account the top and bottom ghost rows for communication with neighboring processes.

### Simulation Loop:

* Implement a loop to perform a fixed number of simulation steps.
* At each step, exchange the top and bottom rows with neighboring processes.
* Update the local grid using the update function.
* Gather the updated local grids to form the full grid on the root process.

### Visualization:

* On the root process (*rank = 0*), print the full grid to the console at each step.

### Testing and Execution:

* Test the script by running it with multiple MPI processes using mpirun command.

## Bonuses:

* Run this program on multiple machines (2 pts).
* Run and correctly display 500x500 grid (2 pts).
* What is the largest grid you can run? The largest grid gets (5 pts).
* Display the evolution of the grid step by step with screen updates (5 pts).