# Task 3:

## Overview:

This Python script is designed to color a grid in such a way that no two adjacent cells (including diagonally adjacent ones) have the same color. It reads input data from files in a directory, performs the coloring algorithm, and writes the results to output files. It demonstrates the use of a simple coloring algorithm in a practical application. The coloring algorithm is widely used in fields like graph theory and computer science for problems such as map coloring and register allocation.

1. ***Importing necessary libraries:*** The script begins by importing necessary libraries such as numpy, pandas, and os.
2. ***Defining helper functions:*** The print\_grid and color\_grid functions are defined. These are used in the main part of the script to perform the coloring algorithm.
   1. print\_grid(grid): This function prints the grid in a readable format.
   2. color\_grid(N, M, grid): This function colors the grid according to the rules. It iterates over each cell in the grid, and if the cell is not an obstacle (denoted by -1), it assigns the smallest available color to it. The available colors for a cell are determined by the get\_available\_colors function, which returns a set of colors that are not used by any of the neighboring cells.
3. ***Reading input files:*** The script reads input files from a directory. Each file contains information about a grid (size, obstacles, etc.). The script processes each file, extracts the necessary information, and constructs the grid.
4. ***Performing the coloring algorithm:*** For each input file, the script performs the coloring algorithm on the corresponding grid.
5. ***Writing output files:*** The script writes the results to output files. Each output file corresponds to an input file and contains the colored grid.

## Now, let’s break down each function:

A black background with white text

Description automatically generated

***print\_grid(grid):*** This function takes a 2D list (grid) as input and prints it row by row. It converts each element of the row to a string and joins them with a space in between. This is a helper function to visualize the grid.

***A screen shot of a computer program

Description automatically generated***

***color\_grid(N, M, grid):*** This function takes the size of the grid (N and M) and the grid itself as input. It defines a nested function get\_available\_colors(x, y) that returns a set of colors that are not used by any of the neighboring cells of the cell at (x, y). Then it iterates over each cell in the grid, and if the cell is not an obstacle (denoted by -1), it assigns the smallest available color to it. The function returns the colored grid.

A screen shot of a computer program

Description automatically generated

The rest of the code outside the functions is the main script that reads the input files, calls the color\_grid function for each one, and writes the results to the output files. It also handles creating the output directory if it doesn’t exist.

The input and output file paths are constructed based on the file names in the input directory and a counter variable.

The grid for each input file is constructed by reading the file line by line and setting the cell values according to the file contents.

After the coloring algorithm is run, the colored grid is written to the output file. The colors are represented by integers, with -1 representing an obstacle. The colors start from 0 and go up to N\*M-1, where N and M are the dimensions of the grid. However, due to the coloring rules, not all colors will be used in the output grid. The colors are assigned in such a way that no two adjacent cells (including diagonally adjacent ones) have the same color. This is ensured by the get\_available\_colors function and the fact that each cell is assigned the smallest available color.