Exercise 4.2

Lucía Martín Fernández

Design (invent!) your own cellular automaton

In [1]:
```
import numpy as np
import random
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
```

In [2]:
```
random.seed(10)
```

My Celullar automata

I used the water weed rule from https://www.complexity-explorables.org/explorables/kelp/ and applied colors so it simulates algae in the sea

In [5]:
```
def cells_toString(cells):
    """
    Transform np.array to string
    """
    str_cells = ''
    for i in range(len(cells)):
        str_cells+=str(int(cells[i]))
    return str_cells
```

In [76…
```
def celullar_automaton(cells):
    """
    Function to get next cell state
    """
    N = len(cells)
    rule = {'000': 0, '100': 0.5, '010': 0.233, '110': 0.863,'001': 0.5, '101': 0.5, '011': 0.653, '111'
    str_cells = cells_toString(cells)

    new_cell = np.zeros(N)
    new_print = np.zeros(N)

    for i in range(N):
        j = i-1
        k = i+1

        if j < 0:
            j = N-1
        if k >= N:
            k = 0

        triple = str_cells[j]+str_cells[i]+str_cells[k]

        for key, value in rule.items():
            if triple == key:
                new_cell[i]=np.random.choice([0,1], p=[1-value, value]) # choose between 1 and 0 dependi
                new_print[i] = value # print the colors

    return new_cell, new_print
```

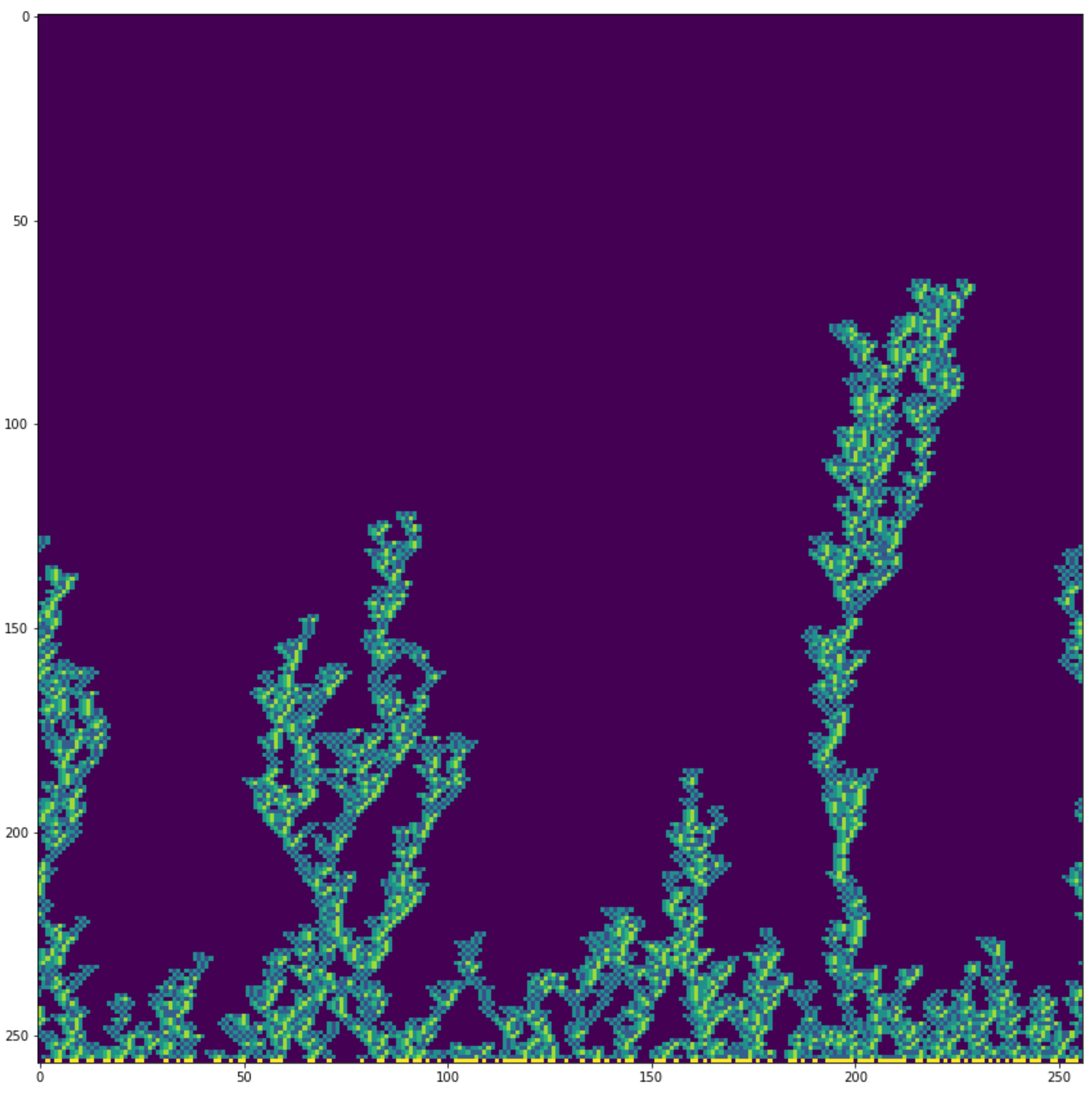
In [11…
```
def plot_cel_automaton(cells):
    """
    Function to plot N_iter of the celullar automaton
    """
    N_iter = 256
    N = len(cells)

    matrix = cells
    last_row = cells
    for i in range(N_iter):
        last_row, last_print = celullar_automaton(last_row)
        matrix = np.vstack([matrix, last_print])

    plt.figure(figsize=(20,15))
    plt.imshow(matrix[::-1],cmap='viridis')
    plt.show()
```

Initial conditions

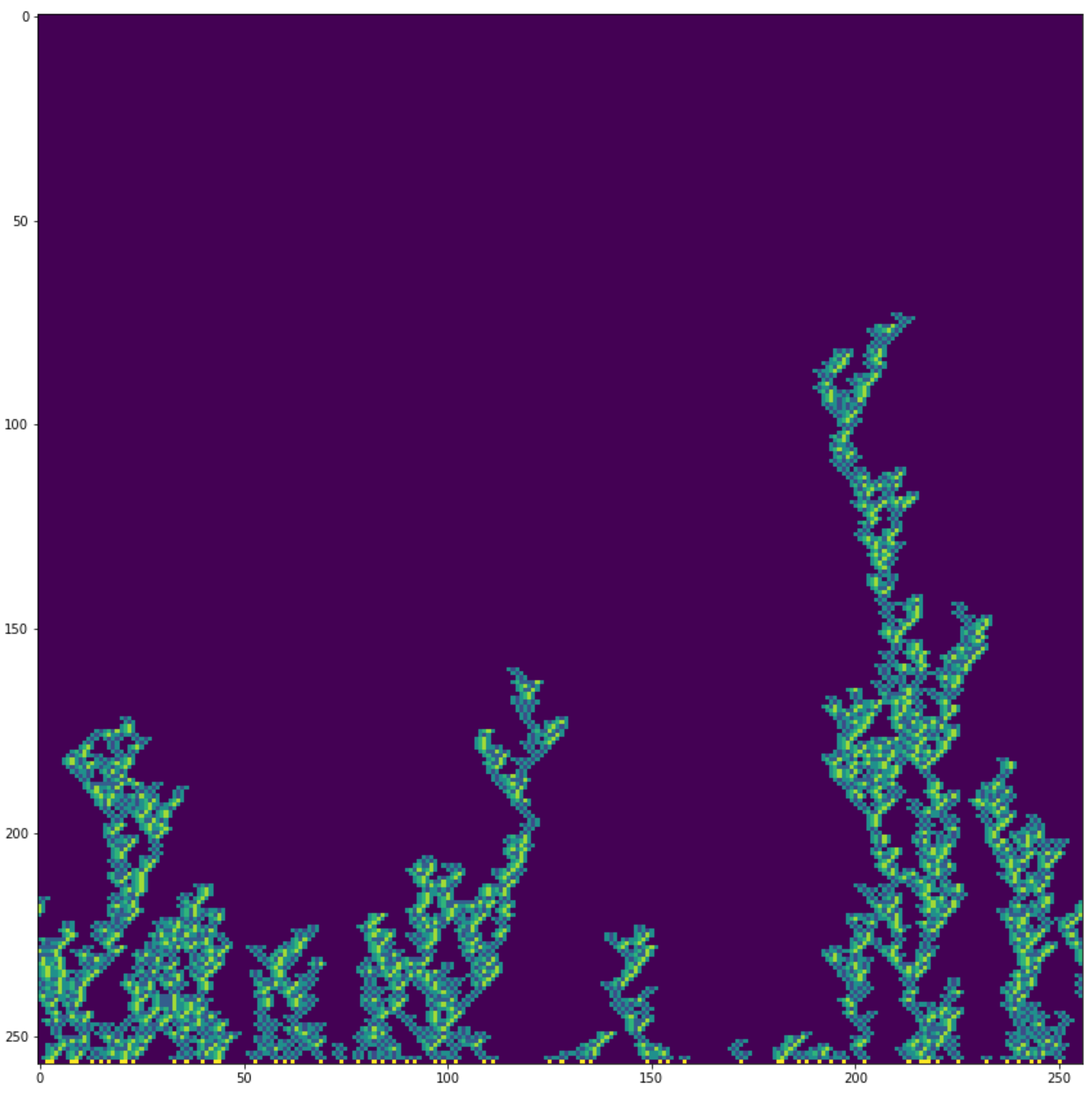
In [11…
```
N =256

# Half black (1) cells and half white (0) cells at randomly chosen positions
cells1 = np.random.randint(2, size=N, dtype=int)

# 25% of black (1) cells and 75% of white (0) cells (approximately)
cells2 = np.random.choice([0, 1], size=N, p=[0.75, 0.25])

# 90% of black (1) cells and 10% of white (0) cells (approximately)
cells3 = np.random.choice([0, 1], size=N, p=[0.1, 0.9])
```

Drawings

In [11…
```
plot_cel_automaton(cells1)
```



In [11…
```
plot_cel_automaton(cells2)
```



In [11…
```
plot_cel_automaton(cells3)
```