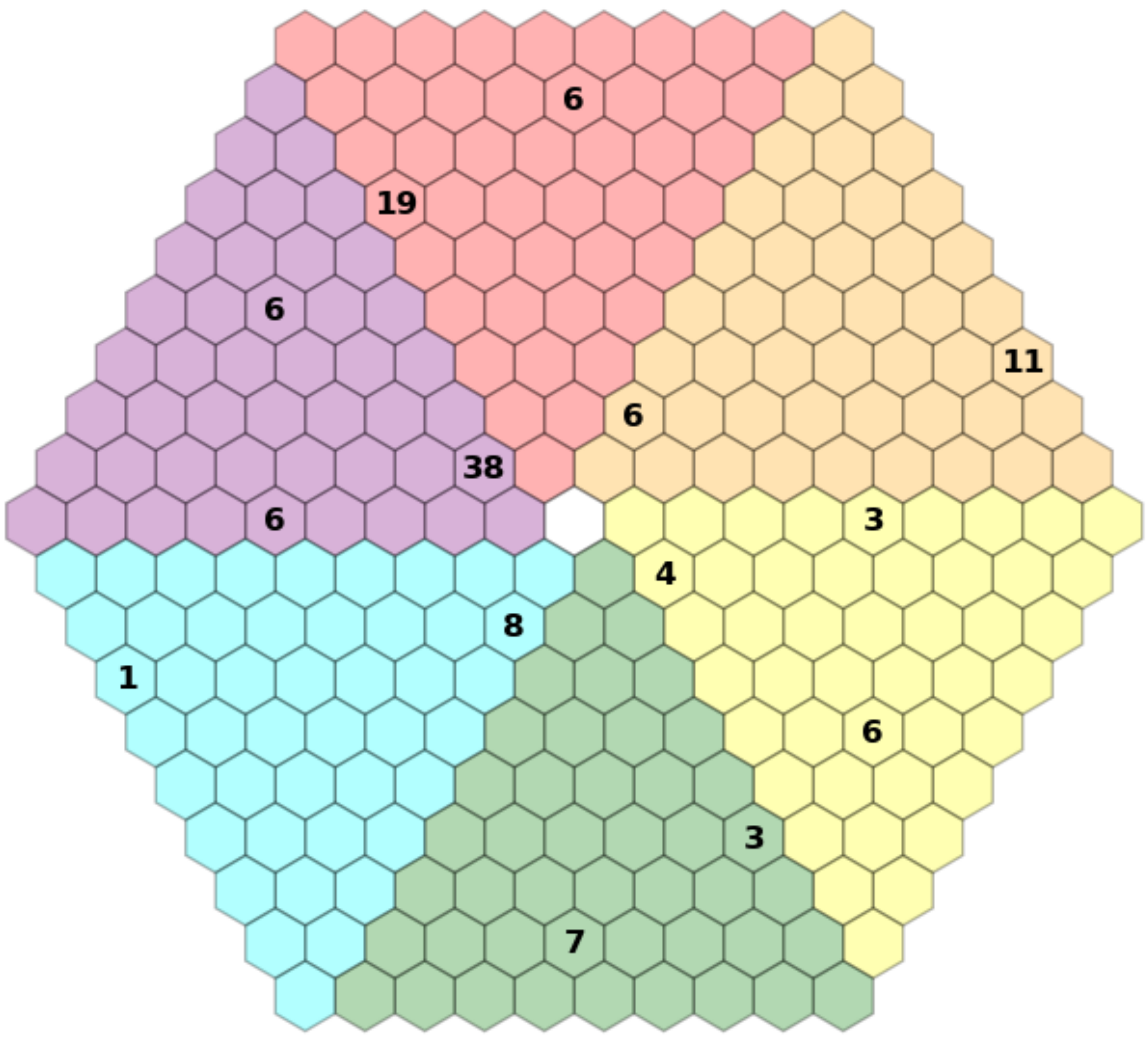
In [1]:

```
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
import math

config = {
    "2-13": 6,
    "3-6": 1,
    "3-15": 19,
    "4-9": 6,
    "9-7": 8,
    "7-10": 38,
    "9-11": 6,
    "5-17": 6,
    "11-8": 4,
    "13-1": 7,
    "15-3": 3,
    "14-9": 3,
    "16-5": 6,
    "15-12": 11,
}

def to_hex(x, y):
    hx = x + y * 0.5
    hy = y * math.sqrt(3) / 2
    return (hx, hy)

def get_color(x, y):
    r1 = x + y > 18
    nr1 = x + y < 18

    r2 = x > 9
    nr2 = x < 9

    r3 = y > 9
    nr3 = y < 9

    colors = [
        ("purple", nr1 and nr2 and not nr3),
        ("cyan", nr1 and not r2 and nr3),
        ("green", not r1 and r2 and nr3),
        ("red", not nr1 and nr2 and r3),
        ("orange", not nr1 and not nr2 and r3),
        ("yellow", r1 and r2 and not r3),
        ("white", True),  # Default color
    ]

    for color, condition in colors:
        if condition:
            return color
    return "white"

def draw_hex_grid_flipped_interactive(width=19, height=19, hex_size=30):
    hex_width = math.floor(hex_size * math.sqrt(3))
    hex_height = math.floor(2 * hex_size)
    vertical_spacing = math.floor(hex_height * 0.75)

    fig, ax = plt.subplots(figsize=(12, 12))
    ax.set_aspect('equal')
    ax.axis('off')

    hexagons = []
    hex_points = [
        (hex_width / 2, 0),
        (hex_width, hex_height / 4),
        (hex_width, (3 * hex_height) / 4),
        (hex_width / 2, hex_height),
        (0, (3 * hex_height) / 4),
        (0, hex_height / 4)
    ]

    total_grid_width = width * hex_width + (height - 1) * (hex_width * 0.5)
    total_grid_height = height * vertical_spacing + hex_height * 0.25

    for y in range(height):
        for x in range(width):
            if y + x < 9 or y + x > 27:
                continue

            flipped_x = width - x - 1
            flipped_y = height - y - 1

            center_x = flipped_x * hex_width + (flipped_y - 9) * hex_width * 0.5 + hex_size
            center_y = flipped_y * vertical_spacing + hex_size

            color = get_color(flipped_x, flipped_y)

            hexagon = Polygon(hex_points, closed=True, facecolor=color, edgecolor='black', alpha=0.3, linewidth=1)
            hexagon.set_xy([
                (px + center_x - hex_width / 2, py + center_y - hex_height / 2)
                for px, py in hex_points
            ])
            ax.add_patch(hexagon)
            hexagons.append((hexagon, (flipped_x, flipped_y)))

            key = f"{flipped_x}-{flipped_y}"
            value = config.get(key, "")
            if value:
                ax.text(center_x, center_y, str(value), ha='center', va='center', fontsize=12, fontweight='bold')

    def on_click(event):
        for hexagon, (hx, hy) in hexagons:
            if hexagon.contains_point([event.x, event.y]):
                key = f"{hx}-{hy}"
                value = config.get(key, "No value")
                print(f"Hexagon clicked at {hx}, {hy} with value: {value}")
                break

    fig.canvas.mpl_connect('button_press_event', on_click)

    ax.set_xlim(-hex_width, total_grid_width + hex_width)
    ax.set_ylim(-hex_height, total_grid_height + hex_height)
    plt.show()

if __name__ == "__main__":
    draw_hex_grid_flipped_interactive()
```



In [ ]: