



### Activity Sheet: Finding the Largest Connected Region in a Binary Matrix

**Objective:** Students will implement a recursive algorithm to find and count connected regions of 1s in a binary matrix. They will follow step-by-step instructions to complete the given code.

#### Part 1: Understanding the Problem

A **binary matrix** consists of 0s and 1s. A **region** is a group of connected 1s that can be connected **horizontally, vertically, or diagonally**.

#### Example Matrix

```
1 1 0 1 1
0 1 0 0 0
0 0 0 1 0
1 0 0 1 0
1 1 1 0 1
```

A function should find the largest region of connected 1s.

#### Part 2: Completing the Function

The function `count_region(matrix, x, y)` should:

1. **Check base conditions:** If  $x, y$  are out of bounds or not 1, return 0.
2. **Mark the cell as visited** by setting it to -1.
3. **Recursively explore all 8 directions.**
4. **Return the size of the connected region.**



### Complete the Missing Code

Fill in the missing parts of the function below:

```
def count_region(matrix, x, y):
```

```
    rows, cols = len(matrix), len(matrix[0])
```

```
    # Step 1: Check base conditions
```

```
    if x < 0 or y < 0 or x >= rows or y >= cols or _____: # (Fill in the condition)
```

```
        return 0
```

```
    # Step 2: Mark as visited
```

```
    matrix[x][y] = -1 # Temporarily mark visited
```

```
    # Step 3: Count this cell + explore all 8 directions
```

```
    size = 1
```

```
    for dx, dy in [(-1, -1), (-1, 0), (-1, 1),
```

```
                  (0, -1),      (0, 1),
```

```
                  (1, -1), (1, 0), (1, 1)]:
```

```
        size += _____ # (Fill in recursive function call)
```

```
    return size
```



### Part 3: Finding the Largest Region

The function `largest_region(matrix)` should:

1. Iterate through each cell in the matrix.
2. If a cell is 1, call `count_region(matrix, i, j)`.
3. Keep track of the maximum region size.

#### Complete the Missing Code

```
def largest_region(matrix):  
    max_region = 0  
    for i in range(len(matrix)):  
        for j in range(len(matrix[0])):  
            if _____: # (Fill in the condition)  
                max_region = max(max_region, count_region(matrix, i, j))  
    return max_region
```



#### Part 4: Running the Function

1. Define the binary matrix.
2. Call `largest_region(matrix)`.
3. Print the largest connected region size.

#### Complete the Code Below

```
binary_matrix = [  
    [1, 1, 0, 1, 1],  
    [0, 1, 0, 0, 0],  
    [0, 0, 0, 1, 0],  
    [1, 0, 0, 1, 0],  
    [1, 1, 1, 0, 1]  
]
```

```
# Find the largest connected region
```

```
print("Largest Region Size:", _____) # (Fill in function call)
```

#### Part 5: Discussion Questions

1. What happens if you change the matrix so that all elements are 1?
2. How would you modify the function to return all connected regions instead of just the largest one?

#### Challenge:

- Modify the function to **restore** the matrix after processing instead of marking visited cells with -1.
- Implement an **iterative version** instead of recursion.