(2)

Input: Binary Image in the form of a 2D matrix consisting of 1’s and 0’s.
Output: Count of objects in the image (or count of islands of adjacent 1’s in the 2D matrix)

Idea:

1. The binary image is represented in the form of 1’s and 0’s. Any two 1’s in the matrix belong to the same object if they are adjacent to each other. If there exists a 0 between two 1’s, then the two 1’s may or may not belong to the same object.
2. We may consider the 1’s as the vertices of a graph. For every white pixel, the neighboring white pixels are connected to it via an undirected unweighted edge.
3. The problem now reduces to finding the number of connected components in the entire image when considered as a graph.
4. We use the BFS technique to solve the problem as follows.

Algorithm:

```
1. Initialize the object_count as 0.
2. Iterate over all the pixels of the image.
    a. If a white pixel is found:-
        i.   Push it to a queue data structure.
        ii.  Increment the object_count by one.
        iii. Repeat while the queue is not empty:-
            1. Pop a pixel from the queue, mark it as black.
            2. Add all the white neighboring pixels to the queue.
3. Return the object_count.
```

Code:

- Attached as a file named:- 2018033_CV_HW1.py
- To run, put the image ‘Project1.png’ in the same directory as the python file.