**Technical Challenge**

We would like you to design an API/service that solves the following problem:

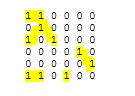
Given a two-dimensional matrix, each of the entries represents a person:

1 0 1

0 0 0

0 1 0

If the entry is 1, we consider the represented person to be sick and we recognise this person as a patient. If two patients are adjacent to each other in any direction (vertical, horizontal and oblique), we consider them part of the same patient group. For example, the following matrix contains 4 patient groups:



Your API will receive a request like the following:

POST /api/patient-groups/calculate HTTP/1.1

Host: {url}:{port}

Content-Length: {content\_length}

content-type: application/json

Accept: \*

{

"matrix" : [

[1, 1, 0, 0, 0, 0],

[0, 1, 0, 0, 0, 0],

[1, 0, 1, 0, 0, 0],

[0, 0, 0, 0, 1, 0],

[0, 0, 0, 0, 0, 1],

[1, 1, 0, 1, 0, 0]

]

}

And return a response body like this:

{

"numberOfGroups": 4

}

Here is another example:

{

"matrix" : [

[1, 0, 1, 1, 1],

[1, 0, 0, 0, 0],

[1, 0, 0, 0, 1],

[0, 0, 1, 0, 0],

[0, 1, 0, 0, 0],

[0, 1, 0, 0, 1]

]

}

And the response body should be:

{

"numberOfGroups": 5

}

Requirements

Please use .NET Core to create this task.

Some things you need to consider are, but not limited to:

1. Code maintainability
2. Memory usage
3. Time complexity
4. Tests

The solution needs to be in a Git repository.

We want to see how you arrived at your solution, so please make your changes in small commits. Any solution that is not in a Git repository or has all the files squashed/added in a single commit will be rejected.

**Please DO NOT post your solution into a public repository such as Github or Bitbucket.**

Please zip the Git repository for your solution and send it back to us.

Good luck!

**Below are couple of best practices(but not limited to) that you can incorporate in this solution.**

1. **Following proper naming conventions in solution, variables, method name, class name etc.**
2. **Pick an optimized solution/algorithm for the problem statement that most suites the problem.**
3. **Ensure that API endpoints, input variable and output variable are correctly defined as mentioned in the problem statement.**
4. **Add comments wherever necessary**
5. **Follow proper folder structures**

**All the Best!**