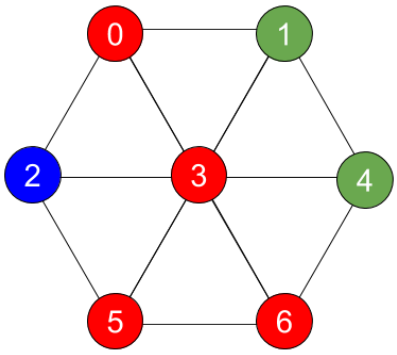
Assignment

Question
Find identically-colored connected components in a triangle mesh with Open3D.

Implement a function to return a list of identically-colored connected components. An identically-colored connected component consists of spatially connected vertices with the same color. In this question, a connected component is represented by a list of vertex indices.

Example triangle mesh



A triangle mesh is represented by vertices and triangles.

In this example, there are 7 vertices:
[0, 1, 2, 3, 4, 5, 6]

There are 6 triangles:
[(0, 2, 3), (0, 3, 1), (1, 3, 4), (2, 5, 3), (3, 5, 6), (3, 6, 4)]

Each vertex has a color:
[red, green, blue, red, green, red, red]

Given this mesh, the expected output of IdenticallyColoredConnectedComponents is:
[[0, 3, 5, 6], [1, 4], [2]]
where the order of the lists are sorted ascendingly by the smallest element in each list; and within each list, the vertices are sorted ascendingly by their indices.

We then write the results into "**results.txt**", where each line in the text file represents one connected component. In each line, vertex indices are separated by space:

```
0 3 5 6
1 4
2
```

Example C++ invocation (solution.cpp)

```
#include "Open3D.h"
using namespace open3d;

int main() {
    // Read triangle mesh "test_mesh.ply"
    geometry::TriangleMesh mesh;
    ...

    // Then get connected components
    auto connected_components = mesh.IdenticallyColoredConnectedComponents();

    // Print connected_components in the specified format
    ...

    return 0;
}
```

Example python invocation (solution.py)

```
import open3d as o3d

# Read triangle mesh "test_mesh.ply"
mesh = ...

# Then get connected components
connected_components = mesh.identically_colored_connected_components()

# Print connected_components in the specified format
```

Tasks

You're expected to:
1. Write C++ function open3d::geometry::TriangleMesh::IdenticallyColoredConnectedComponents
2. Write the Python binding for open3d.geometry.TriangleMesh.identically_colored_connected_components
3. Write a "**Open3D/examples/Cpp/solution.cpp**" file to read the input mesh "test_mesh.ply", find identically-colored connected components. Change the build system so that an executable can be build
4. Write a "**Open3D/examples/Python/Basic/solution.py**" file to read the input mesh "test_mesh.ply", find identically-colored connected components
5. Output the result of task 3 or 4 to "**results.txt**" (their results shall be the same), where the results shall be formatted the same way as specified in the "Example triangle mesh" section
6. Write C++ and Python unit tests integrated with Open3D's unit test system
7. Document your code, the algorithm used, how to build and run, and etc.

Submission

To submit your code: push the code to a **private** git repository, and let us know the access method to the repository. Include all the necessary files, including "results.txt" in your repository.

Useful links

- Open3D repository: https://github.com/intel-isl/Open3D
- Open3D docs: http://www.open3d.org/docs
- Pybind11: https://github.com/pybind/pybind11