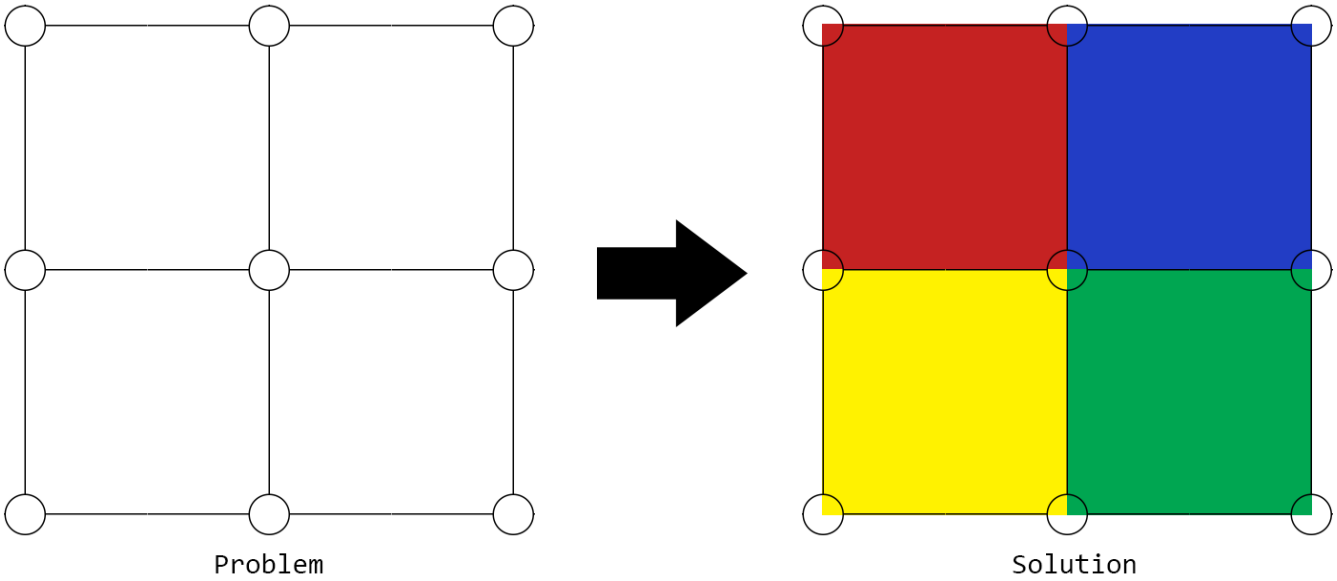
You are tasked with solving the following procedural generation problem in Unity3D:

Given a graph of Points and Edges like the one pictured below, find and create the resulting 3D meshes that will fill all empty areas within the following criteria:

* No two meshes overlap
* No edges overlap the center of a mesh
* The meshes' boundary edges always overlap with a graph edge
* Each mesh will render in a way that can be seen from a consistent view angle (Y-up)



The only solution to the problem on the left is four square meshes that each fill a quarter of the graph.

Included in this package is a GIF of the result we're looking for and a test project containing four problems. Even if you can't pass every test within the time limit, we want to see your result.

What we are looking for:

* A result that matches the GIF provided (material differentiation is optional)
* An optimized solution that achieves as many frames per second as you can achieve
* A flexible solution that works for every test. No hard references
* A full 3D solution. No 2D screenspace trickery
* No two-sided materials
* Proper use of data structures and algorithms
* Detailed commenting, describing what you're doing and why

Note that the "TestSource" directory will be completely overwritten when we assess the end result of your test. Any changes made to the prefabs or source code within will be deleted.

We used Unity 2017.2.2f1 to produce this test, but the code and assets should work in late 5.x builds.