Mega Assignments

I'll explain all in my Friday (Feb 27) class.

1. **Orthogonal piping:** Given a sequence S of voxels (3D integer points) as input. Join them in maximal-length sub-sequences as orthogonally flowing pipes of square cross-section with section dimension d, specified by the user. Two consecutive voxels of S will be consecutive and belong to the same pipe (i.e., sub-sequence) if and only if they have a common face. The final output should be an obj file in which each pipe should be defined as a set of minimum number of faces, each face being an orthogonal polygon. Suggested data structure: DCEL. The final rendition would look something like http://www.bulcy.com/zwayFiles/Houdini/network_12.1.jpg

2. **Orthogonal slicing:** Given a 3D point cloud, construct its three sets of orthogonal slice polygons along three coordinate axes. Save each set of polygons as an OBJ file.

3. **Vox-to-DCEL:** Use DCEL to get a compact representation of a voxel set. In this rep, each face would be a maximum-area orthogonal polygon with axis-parallel edges.

4. **3D fractals:** http://www.sciencedirect.com/science/article/pii/S0097849308000964

5. **Randomized tiling:** Randomized sub-optimal tiling of a 2D digital object by a suitable square tile denomination.

6. **OBJ-to-SVG:** Rendition of an OBJ file to produce an SVG output after perspective projection, hidden surface removal, and illumination.

7. **Spherical voxels:** From a 3D point set to SVG. See http://www.designzzz.com/wp-content/uploads/2011/01/lattice-modifier.jpg

8. **Quad cover:** Give a 2D digital object A, use (alternately orthogonal) median lines to partition A. And then construct its quad-cover as an orthogonal polygon. An example: https://www.staff.ncl.ac.uk/qiuhua.liang/Research/grid_generation_files/image002.gif

9. SVG animation: http://www.petercollingridge.co.uk/sites/files/peter/4gears.svg

10. Tangram game animation in OpenGL or Java.

    * Last two are of low potential, hence not recommended for an ambitious student :-)