```
input  : a collection of edges, and a collection of vertexes
output: resolved grid point to grid point visibility
for each grid point in ever edge do
    pick one random vertex and cast a ray to that vertex;
    cast rays to all the other vertexes;
    sort other vertexes by the angle between that one vertex ray, and
    other vertexes rays;
    prepare an array to hold a slice of an edge, same size as number of
    vertexes;
    for each non transparent edge, but the one the grid point belongs
    to do
        find its end vertexes positions in the sorted array, loop around
        if necessary (the two vertexes this edge connects);
        divide the edge into edge slices using rays between this edge
        end vertexes, as determined by vertexes index in sorted array;
        for for each sliced edge piece do
            if array is empty at slicing ray vertex index or this edge
            slice is closer to grid point than the one already occupying
            the array then
                put edge slice in the structure at slicing ray vertex
                index;
            end
        end
    end
    for each edge slice in the array do
        mark all coressponding points as visible by the grid point
        considered in outer loop;
    end
end
```