extract the collaboration subgraph consisting of all the professors from CSE

```
var cseID = rawData.nodes.filter(e => e.dept === "CSE").map(e => e.id);
            var data = {
                nodes: rawData.nodes.filter(e => e.dept === "CSE"),
                edges: rawData.edges.filter(e => cseID.indexOf(e.source
) >= 0 && cseID.indexOf(e.target) >= 0)
            }
            var noNameCount = 0
            data.nodes.map(e => {
                if (!e.fullname) {
                    e.fullname = "No Name " + noNameCount;
                    noNameCount++;
                }
                e.collaborators = data.edges.filter(f => f.source === e
.id || f.target === e.id).map(f => {
                    var cObj = {};
                    if (f.target === e.id) {
                        cObj.id = f.source;
                    }
                    else {
                        cObj.id = f.target;
                    }
                    cObj.publications = f.publications;
                    return cObj;
                });
            })
```

get the degree of each vertex

```
var edge_num = {}
            for (i = 0; i < data.nodes.length; i++) {
                    edge_num[data.nodes[i].id] =1
                }

            for (i = 0; i < data.edges.length; i++) {
                    edge_num[data.edges[i].source] = edge_num[data.edge
s[i].source] + 1
                    edge_num[data.edges[i].target] = edge_num[data.edge
s[i].target] + 1
                }
```

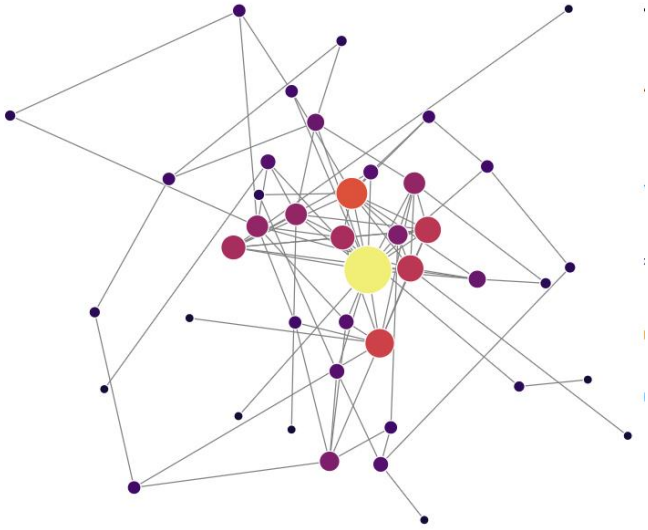
Sets the properties of vertices and edges

```
var link = g.append("g")
                .attr("class", "links")
                .selectAll("line")
```

```
                .data(data.edges)
                .enter().append("line")
                .attr("stroke", "grey")
                .attr("stroke-width", 1);

            var node = g.append("g")
                .attr("class", "nodes")
                .selectAll("g")
                .data(data.nodes)
                .enter().append("g")

            var circles = node.append("circle")
                .attr("r", function(d) { return d.collaborators.length
+ 3; })
                .attr("fill", function(d) {
                    node_id = d.id
                    r = edge_num[node_id]
                    return myColor(r)
                } )
                .attr("stroke", "white")
                .attr("stroke-width", 1)
                .on("mouseover", nodeLinkHighlight())
                .on("mouseout", nodeLinkReset);
```

Map your social network

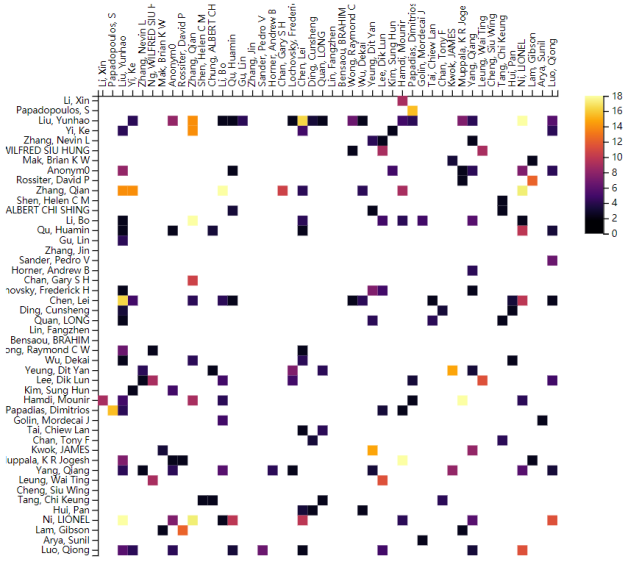
```
simulation
                .nodes(data.nodes)
                .on("tick", ticked);

            simulation.force("link")
                .links(data.edges);

            function ticked() {
                link
                    .attr("x1", function(d) { return d.source.x; })
                    .attr("y1", function(d) { return d.source.y; })
                    .attr("x2", function(d) { return d.target.x; })
                    .attr("y2", function(d) { return d.target.y; });

                node
                    .attr("transform", function(d) {
                        return "translate(" + d.x + "," + d.y + ")";
                    })
            }
```



Draw a matrix heat map

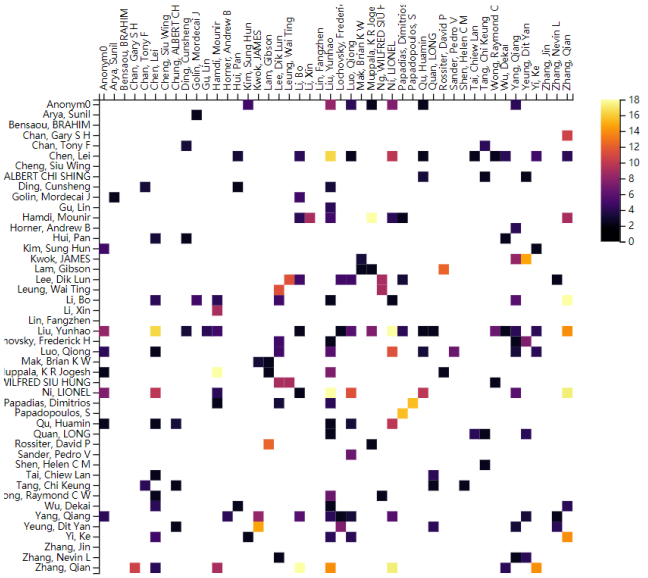
Sets the properties of the small rectangle

```
var rects = m_GG.selectAll("rect")
                .data(d => { d.neigh.forEach(e => e.source = d.id); ret
urn d.neigh;})
                .enter()
                .append("rect")
                .attr("x", d => idScale(d.id))
                .attr("y", d => idScale(d.source))
                .attr("width", nameScale.bandwidth())
                .attr("height", nameScale.bandwidth())
                .attr("fill", d => myColor(d.publications.length))
                .on("mouseover", matrixHighlight())
                .on("mouseout", matrixReset);
```



Sort the matrix

```
var sort_matrix = d3.select("button#sort").on("click", function() {
                console.log("sort by name")
                data.nodes.sort(sortByName);
                // Update axis
                nameScale.domain(data.nodes.map(e => e.fullname)).range
([0, m_Width]);
                idScale.domain(data.nodes.map(e => e.id)).range([0, m_W
idth]);

                x_axis.transition().duration(750).delay((d, i) => i * 2
0).call(d3.axisTop(nameScale)).selectAll("text").style("text-
anchor", "start").attr("transform", "translate(12,-10) rotate(-90)");
                y_axis.transition().duration(750).delay((d, i) => i * 2
0).call(d3.axisLeft(nameScale));
                rects
                    .order()
                    .transition()
                    .duration(750)
                    .delay((d, i) => i * 20)
                    .attr("x", d => idScale(d.id))
                    .attr("y", d => idScale(d.source));
            });

function sort_name(a, b) {
                return d3.ascending(a.fullname, b.fullname);
            }
```



Linkage: when hovering the mouse on node in the node-link diagram, the corresponding column and row of the matrix view should be highlighted

```
function NLHighlight() {
                return function(d) {
                    rects.style("opacity", e => e.id === d.id || e.sour
ce === d.id ? 1 : 0);
                    node.style("stroke-opacity", function(e) {
                        if (e.id === d.id) {
                            return 1;
                        }
                        return d.neigh.find(function(f) { return f.id =
== e.id }) === undefined ? 0.2 : 1;
                    });
                    node.style("fill-opacity", function(e) {
                        if (e.id === d.id) {
                            return 1;
                        }
                        return d.neigh.find(function(f) { return f.id =
== e.id }) === undefined ? 0.2 : 1;
                    });
                    link.style("stroke-opacity", function(e) {
                        return e.source.id === d.id || e.target.id ===
d.id ? 1 : 0.2;
                    });
                }
            }
```

when hovering on a cell in the matrix view, the corresponding nodes and edges should be highlighted as well.

```
function m_Highlight() {
                return function(d) {
                    rects.style("opacity", e => e.id === d.id && e.sour
ce === d.source ? 1 : 0);
                    node.style("stroke-opacity", function(e) {
                        if (e.id === d.id || e.id === d.source) {
                            return 1;
                        }
                        return 0.2;
                    });
                    node.style("fill-opacity", function(e) {
                        if (e.id === d.id || e.id === d.source) {
                            return 1;
                        }
                        return 0.2;
```

```
                    });
                    link.style("stroke-opacity", function(e) {
                        return e.source.id === d.id && e.target.id ===
d.source || e.source.id === d.source && e.target.id === d.id ? 1 : 0.2;
                    });
                    link.style("stroke-width", function(e) {
                        return e.source.id === d.id && e.target.id ===
d.source || e.source.id === d.source && e.target.id === d.id ? 1 + e.pu
blications.length : 1;
                    });
                }
            }
```