(function() {

/\*\*

\* @namespace randomgraph

\*/

var randomgraph = {

/\*\*

\* Simple balanced tree

\*

\* @memberof randomgraph

\* @param {Number} r number of children each node has

\* @param {Number} h height of the tree

\*/

BalancedTree: function(r, h) {

var v = 0,

graph = { nodes: [{ label: 'node 0', index: 0 }], edges: [] },

newLeaves = [],

i, j, height, node, leaves;

for (i = 0; i < r; i++) {

node = { label: 'node '+(++v), index: (v-1) };

graph.nodes.push(node);

newLeaves.push(node);

graph.edges.push({ source: 0, target: v });

}

for (height = 1; height < h; height++) {

leaves = newLeaves;

newLeaves = [];

for (j = 0; j < leaves.length; j++) {

for (i = 0; i < r; i++) {

node = { label: 'node '+(++v), index: (v-1) };

newLeaves.push(node);

graph.nodes.push(node);

graph.edges.push({ source: leaves[j].index, target: v });

}

}

}

return graph;

},

/\*\*

\* @namespace randomgraph.ErdosRenyi

\*/

ErdosRenyi: {

/\*\*

\* Erdos–Rényi aka Gilbert

\*

\* @memberof randomgraph.ErdosRenyi

\* @param {Number} n number of nodes

\* @param {Number} p probability of a edge between any two nodes

\*/

np: function(n, p) {

var graph = { nodes: [], edges: [] },

i, j;

for (i = 0; i < n; i++) {

graph.nodes.push({ label: 'node '+i });

for (j = 0; j < i; j++) {

if (Math.random() < p) {

graph.edges.push({

source: i,

target: j

});

}

}

}

return graph;

}

}

};

// CommonJS module is defined

if (typeof module !== 'undefined' && module.exports) {

module.exports = randomgraph;

}

/\*global ender:false \*/

if (typeof ender === 'undefined') {

// here, `this` means `window` in the browser, or `global` on the server

// add `numeral` as a global object via a string identifier,

// for Closure Compiler 'advanced' mode

this['randomgraph'] = randomgraph;

}

/\*global define:false \*/

if (typeof define === 'function' && define.amd) {

define([], function () {

return randomgraph;

});

}

})();