

Vizualizacija podataka - Laboratorijska vježba 4

Prikazi

1. Prvi primjer iz ovog predloška potrebno je modificirati kako bi se ispisalo i vrstu voća zajedno s količinom.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <title>Prvi zadatak</title>
  </head>
  <body>
    <script>
      var data = [
        {name: 'jabuka', value: 17},
        {name: 'kruska', value: 4},
        {name: 'banana', value: 83},
        {name: 'jagoda', value: 47},
        {name: 'ribizla', value: 1}
      ];

      var width = 500;
      var height = 500;
      var outerRadius = 200;
      var innerRadius = 0;

      var color = d3.scale.category20();

      var arc = d3.svg.arc()
        .innerRadius(innerRadius)
        .outerRadius(outerRadius);

      var pie = d3.layout.pie()
        .value(function(d) { return d.value; });

      var svg = d3.select("body")
        .append("svg")
        .attr("width", width)
        .attr("height", height);

      var pieArcs = svg.selectAll("g.pie")
        .data(pie(data))
        .enter()
        .append("g")
        .attr("class", "pie")
```

```

.attr("transform", "translate(" + (width / 2) + ", " + (height / 2) + ")");

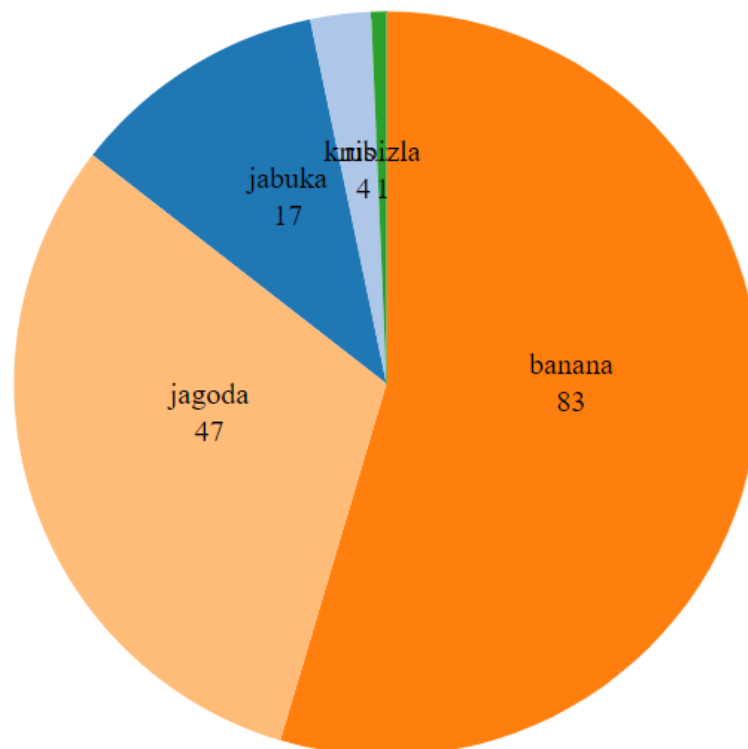
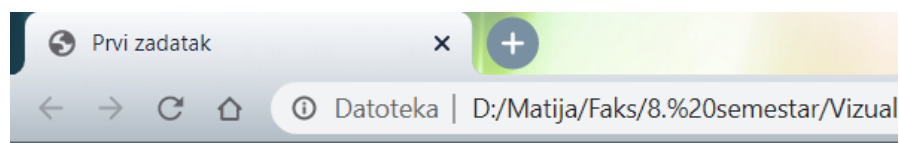
pieArcs.append("path")
.attr("fill", function(d, i) { return color(i); })
.attr("d", arc);

pieArcs.append("text")
.attr("transform", function(d) { return "translate(" + arc.centroid(d) + ")"; })
.attr("text-anchor", "middle")
.text(function(d) { return d.value; });

pieArcs.append("text")
.attr("transform", function(d) { return "translate(" + arc.centroid(d) + ")"; })
.attr("text-anchor", "middle")
.attr("baseline-shift", "20px")
.text(function(d, i) { return data[i].name; });
</script>
</body>
</html>

```

Izlistanje koda 1. Sadržaj Z4-1.html datoteke



Slika 1. Rješenje 1. zadatka

Komentar:

U izradi ovoga zadatka korišten je primjer iz predloška koji je promijenjen kako bi se prikazalo ispravno stanje. Uz ostale promjene, bilo je potrebno pomaknuti gore ili dolje jedan od tekstova kako se isti nebi preklapali. Rješenje je prikazano na 1. slici.

- 2. Potrebno je izmijeniti vrijednosti za količinu voća kako bi se promotriilo kako se iscrtavaju elementi vizualizacije u smislu njihovog položaja (što je ispod/iznad čega). Nakon što je nepravilnost uočena, potrebno ju je ispraviti na dva načina.**

a) Postaviti sve tekstove iznad kružnog prikaza.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <title>Drugi(a) zadatak</title>
  </head>
  <body>
    <script>
      var data = [
        {name: 'jabuka', value: 17},
        {name: 'kruska', value: 4},
        {name: 'banana', value: 83},
        {name: 'jagoda', value: 47},
        {name: 'ribizla', value: 1}
      ];

      var width = 500;
      var height = 500;
      var outerRadius = 200;
      var innerRadius = 50;

      var color = d3.scale.category20();

      var arc = d3.svg.arc()
        .outerRadius(outerRadius);

      var pie = d3.layout.pie()
        .value(function(d) { return d.value; });

      var svg = d3.select("body")
        .append("svg")
        .attr("width", width)
        .attr("height", height);

      var pieArcs = svg.selectAll("g.pie")
        .data(pie(data))
```

```

        .enter()
        .append("g")
        .attr("class", "pie")
        .attr("transform", "translate(" + (width / 2) + ", " + (height / 2) + ")");

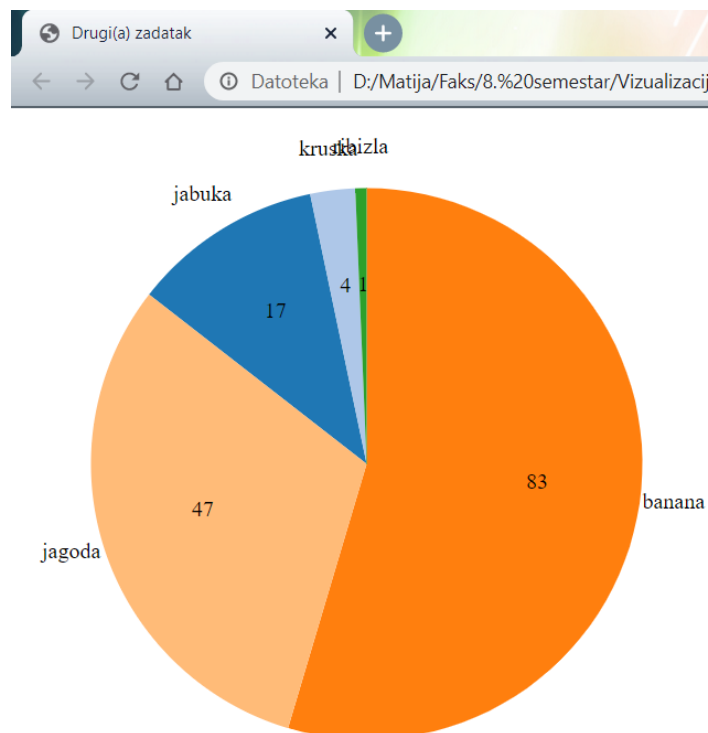
    pieArcs.append("path")
        .attr("fill", function(d, i) { return color(i); })
        .attr("d", arc);

    pieArcs.append("text")
        .attr("transform", function(d) {
            d.outerRadius = outerRadius;
            d.innerRadius = innerRadius;
            return "translate(" + arc.centroid(d) + ")"; })
        .attr("text-anchor", "middle")
        .text(function(d) { return d.value; });

    pieArcs.append("text")
        .attr("transform", function(d) {
            d.outerRadius = outerRadius + 50;
            d.innerRadius = outerRadius + 50;
            return "translate(" + arc.centroid(d) + ")"; })
        .attr("text-anchor", "middle")
        .text(function(d, i) { return data[i].name; });
    </script>
</body>
</html>

```

Izlistanje koda 2. Sadržaj Z4-2a.html datoteke



Slika 2. Rješenje 2. a) zadatka

b) Rotirati tekst kako bi se u manjoj mjeri preklapao s tekstom susjednih segmenata.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <title>Drugi(b) zadatak</title>
  </head>
  <body>
    <script>
      var data = [
        {name: 'jabuka', value: 17},
        {name: 'kruska', value: 4},
        {name: 'banana', value: 83},
        {name: 'jagoda', value: 47},
        {name: 'ribizla', value: 1}
      ];

      var width = 500;
      var height = 500;
      var outerRadius = 200;
      var innerRadius = 50;

      var color = d3.scale.category20();

      var arc = d3.svg.arc()
        .outerRadius(outerRadius);

      var pie = d3.layout.pie()
        .value(function(d) { return d.value; });

      var svg = d3.select("body")
        .append("svg")
        .attr("width", width)
        .attr("height", height);

      var pieArcs = svg.selectAll("g.pie")
        .data(pie(data))
        .enter()
        .append("g")
        .attr("class", "pie")
        .attr("transform", "translate(" + (width / 2) + ", " + (height / 2) + ")");

      pieArcs.append("path")
        .attr("fill", function(d, i) { return color(i); })
        .attr("d", arc);

      pieArcs.append("text")
        .attr("transform", function(d) {
```

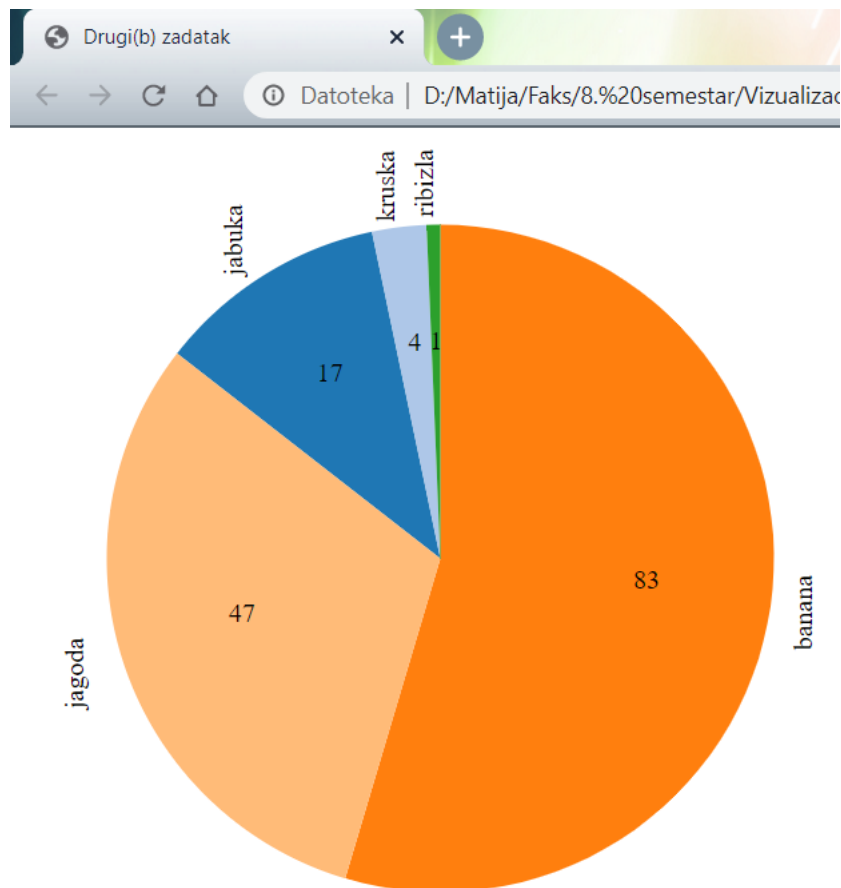
```

        d.outerRadius = outerRadius;
        d.innerRadius = innerRadius;
        return "translate(" + arc.centroid(d) + ")"; })
    .attr("text-anchor", "middle")
    .text(function(d) { return d.value; });

    pieArcs.append("text")
    .attr("transform", function(d) {
        d.outerRadius = outerRadius + 50;
        d.innerRadius = outerRadius + 50;
        return "translate(" + arc.centroid(d) + ") rotate(" + 270 + ")"; })
    .attr("text-anchor", "middle")
    .text(function(d, i) { return data[i].name; });
</script>
</body>
</html>

```

Izlistanje koda 3. Sadržaj Z4-2b.html datoteke



Slika 3. Rješenje 2. b) zadatka

Komentar:

Kako bi se nazivi voća prikazali iznad kružnog prikaza, povećana je vrijednost za unutrašnji radijus prilikom dodavanja novog tekstualnog elementa. U drugom primjeru odrađeno je translaticiranje, tj. rotiranje kako se tekstovi nebi preklapali. Rješenja su prikazana na 2. i 3. slici.

3. Registrirati odgovarajuće događaje (engl. event) za pojedine segmente kružnog prikaza koji će ispisati u konzolnom prozoru naziv i količinu.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <title>Treći zadatak</title>
  </head>
  <body>
    <script>
      var data = [
        {name: 'jabuka', value: 17},
        {name: 'kruska', value: 4},
        {name: 'banana', value: 83},
        {name: 'jagoda', value: 47},
        {name: 'ribizla', value: 1}
      ];

      var width = 500;
      var height = 500;
      var outerRadius = 200;
      var innerRadius = 50;

      var color = d3.scale.category20();

      var arc = d3.svg.arc()
        .outerRadius(outerRadius);

      var pie = d3.layout.pie()
        .value(function(d) { return d.value; });

      var svg = d3.select("body")
        .append("svg")
        .attr("width", width)
        .attr("height", height);

      var pieArcs = svg.selectAll("g.pie")
        .data(pie(data))
        .enter()
        .append("g")
        .attr("class", "pie")
        .attr("transform", "translate(" + (width / 2) + ", " + (height / 2) + ")")
        .on("click", function(d, i) { return console.log('Naziv: ${data[i].name} Kolicina: ${d.value}');
      });

      pieArcs.append("path")
        .attr("fill", function(d, i) { return color(i); })
        .attr("d", arc);
```

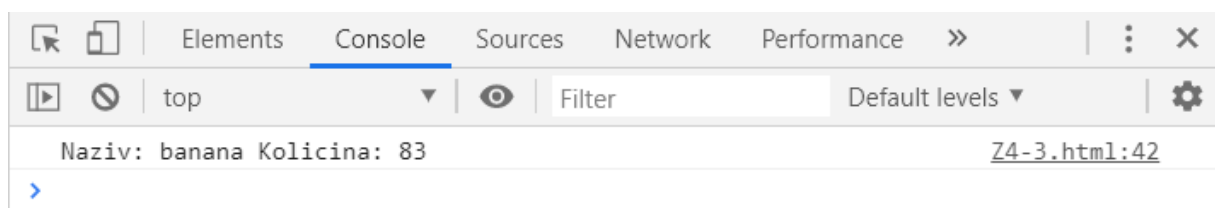
```

    pieArcs.append("text")
    .attr("transform", function(d) {
        d.outerRadius = outerRadius;
        d.innerRadius = innerRadius;
        return "translate(" + arc.centroid(d) + ")"; })
    .attr("text-anchor", "middle")
    .text(function(d) { return d.value; });

    pieArcs.append("text")
    .attr("transform", function(d) {
        d.outerRadius = outerRadius + 50;
        d.innerRadius = outerRadius + 50;
        return "translate(" + arc.centroid(d) + ") rotate(" + 270 + ")"; })
    .attr("text-anchor", "middle")
    .text(function(d, i) { return data[i].name; });
</script>
</body>
</html>

```

Izlistanje koda 4. Sadržaj Z4-3.html datoteke



Slika 4. Rješenje 3. zadatka

Komentar:

Prilikom dodavanja svakog elementa kružnog prikaza dodana je metoda koja se okida prilikom klika na kružni isječak. Za ispis u konzolu korištena je *log()* metoda na *console* objektu koja kao parametar prima vrijednost koja će biti ispisana u konzolu. Rješenje zadatka je prikazano na 4. slici.

4. Izraditi stablo i doraditi kod kako bi se omogućilo skrivanje čvorova djece čvorova kojega korisnik odabere.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://d3js.org/d3.v3.min.js"></script>
    <title>Četvrti zadatak</title>

    <style>
      .node {

```



```

        cursor: pointer;
    }

    .node circle {
        fill: #fff;
        stroke: steelblue;
        stroke-width: 1.5px;
    }

    .node {
        font: 10px sans-serif;
    }

    .link {
        fill: none;
        stroke: #ccc;
        stroke-width: 1.5px;
    }
</style>
</head>
<body>
<script>
    var data = [{
        "name": "flare",
        "children": [
            {
                "name": "analytics",
                "children": [
                    {
                        "name": "cluster",
                        "children": [
                            {"name": "AgglomerativeCluster", "size": 3938},
                            {"name": "CommunityStructure", "size": 3812},
                            {"name": "MergeEdge", "size": 743}
                        ]
                    },
                    {
                        "name": "graph",
                        "children": [
                            {"name": "BetweennessCentrality", "size": 3534},
                            {"name": "LinkDistance", "size": 5731}
                        ]
                    }
                ]
            }
        ]
    }
    ];

    var margin = {top: 20, right: 50, bottom: 20, left: 50};
    var width = 750, height = 750;

```

```

var i = 0, duration = 800;

var tree = d3.layout.tree()
    .size([height, width]);

var diagonal = d3.svg.diagonal()
    .projection(function(d) { return [d.y, d.x]; });

var svg = d3.select("body").append("svg")
    .attr("width", width + margin.right + margin.left)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top +
    ")");

var root = data[0];
root.x0 = height / 2;
root.y0 = 0;

function collapse(d) {
    if (d.children) {
        d._children = d.children;
        d._children.forEach(collapse);
        d.children = null;
    }
}

root.children.forEach(collapse);
update(root);

d3.select(self.frameElement).style("height", "800px");

function update(source) {
    // Compute the new tree layout.
    var nodes = tree.nodes(root).reverse(),
        links = tree.links(nodes);

    // Normalize for fixed-depth.
    nodes.forEach(function(d) { d.y = d.depth * 180; });

    // Update the nodes...
    var node = svg.selectAll("g.node")
        .data(nodes, function(d) { return d.id || (d.id = ++i); });

    // Enter any new nodes at the parent's previous position.
    var nodeEnter = node.enter().append("g")
        .attr("class", "node")
        .attr("transform", function(d) { return "translate(" + source.y0 + "," + source.x0 + ")"; })
        .on("click", click);

    nodeEnter.append("circle")
        .attr("r", 1e-6)

```

```

        .style("fill", function(d) { return d._children ? "lightsteelblue" : "#fff"; });

nodeEnter.append("text")
    .attr("x", function(d) { return d.children || d._children ? -10 : 10; })
    .attr("dy", ".35em")
    .attr("text-anchor", function(d) { return d.children || d._children ? "end" : "start"; })
    .text(function(d) { return d.name; })
    .style("fill-opacity", 1e-6);

// Transition nodes to their new position.
var nodeUpdate = node.transition()
    .duration(duration)
    .attr("transform", function(d) { return "translate(" + d.y + "," + d.x + ")"; });

nodeUpdate.select("circle")
    .attr("r", 4.5)
    .style("fill", function(d) { return d._children ? "lightsteelblue" : "#fff"; });

nodeUpdate.select("text")
    .style("fill-opacity", 1);

// Transition exiting nodes to the parent's new position.
var nodeExit = node.exit().transition()
    .duration(duration)
    .attr("transform", function(d) { return "translate(" + source.y + "," + source.x + ")"; })
    .remove();

nodeExit.select("circle")
    .attr("r", 1e-6);

nodeExit.select("text")
    .style("fill-opacity", 1e-6);

// Update the links...
var link = svg.selectAll("path.link")
    .data(links, function(d) { return d.target.id; });

// Enter any new links at the parent's previous position.
link.enter().insert("path", "g")
    .attr("class", "link")
    .attr("d", function(d) {
        var o = {x: source.x0, y: source.y0};
        return diagonal({source: o, target: o});
    });

// Transition links to their new position.
link.transition()
    .duration(duration)
    .attr("d", diagonal);

// Transition exiting nodes to the parent's new position.
link.exit().transition()

```

```

        .duration(duration)
        .attr("d", function(d) {
            var o = {x: source.x, y: source.y};
            return diagonal({source: o, target: o});
        })
        .remove();

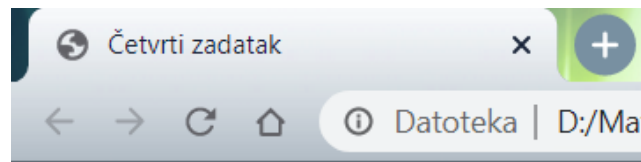
    // Stash the old positions for transition.
    nodes.forEach(function(d) {
        d.x0 = d.x;
        d.y0 = d.y;
    });
}

function click(d) {
    if (d.children) {
        d._children = d.children;
        d.children = null;
    } else {
        d.children = d._children;
        d._children = null;
    }
    update(d)
}

</script>
</body>
</html>

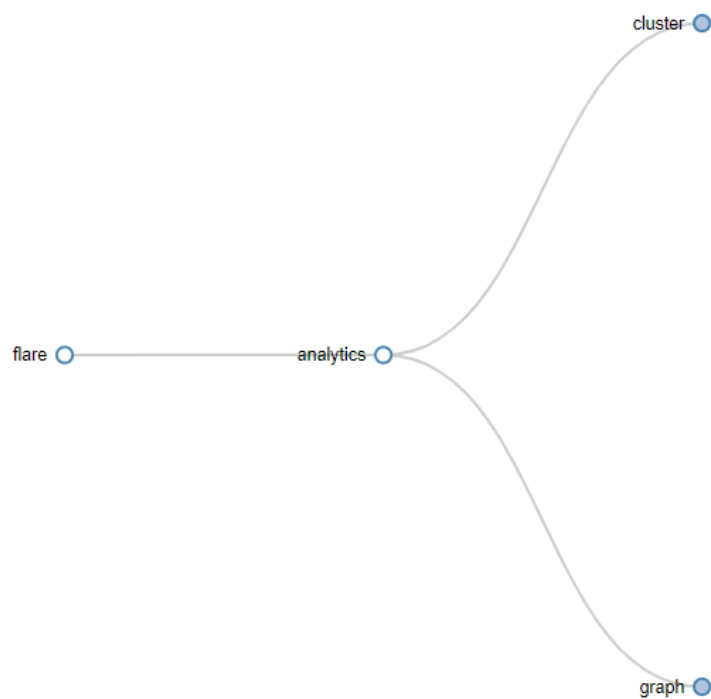
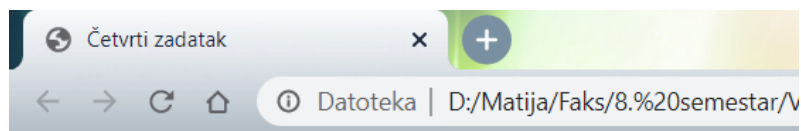
```

Izlistanje koda 5. Sadržaj Z4-4.html datoteke

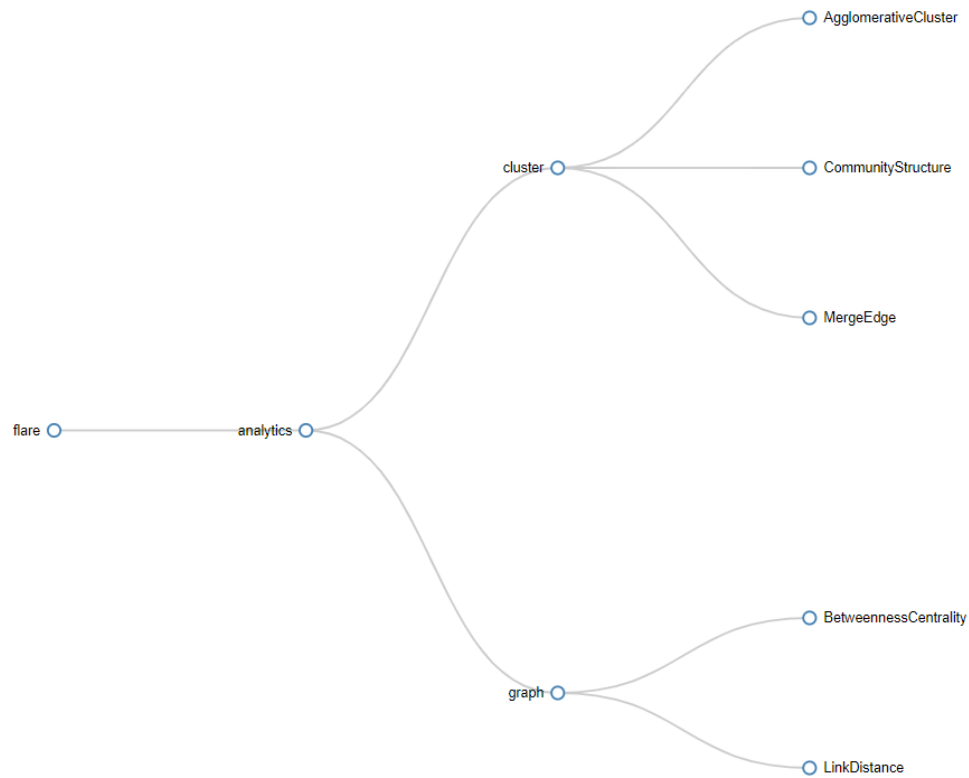
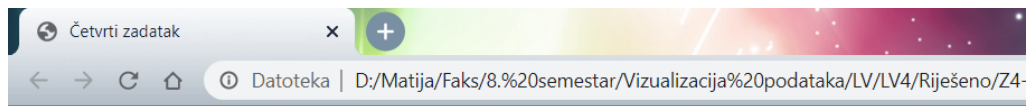


flare ☐ ————— ☒ analytics

Slika 5. Početno stanje 4. zadatka



Slika 6. Stanje nakon klika na „analytics“ element



Slika 7. Stanje nakon svih otvorenih elemenata

Komentar:

Pri rješavanju ovoga zadatka korišteni su razni izvori na internetu kao i primjeri u predlošku za laboratorijsku vježbu. Budući da je kod poprilično složen, dijelovi koda su objašnjeni komentarima. Rješenja zadatka su prikazana na 5., 6. i 7. slici.