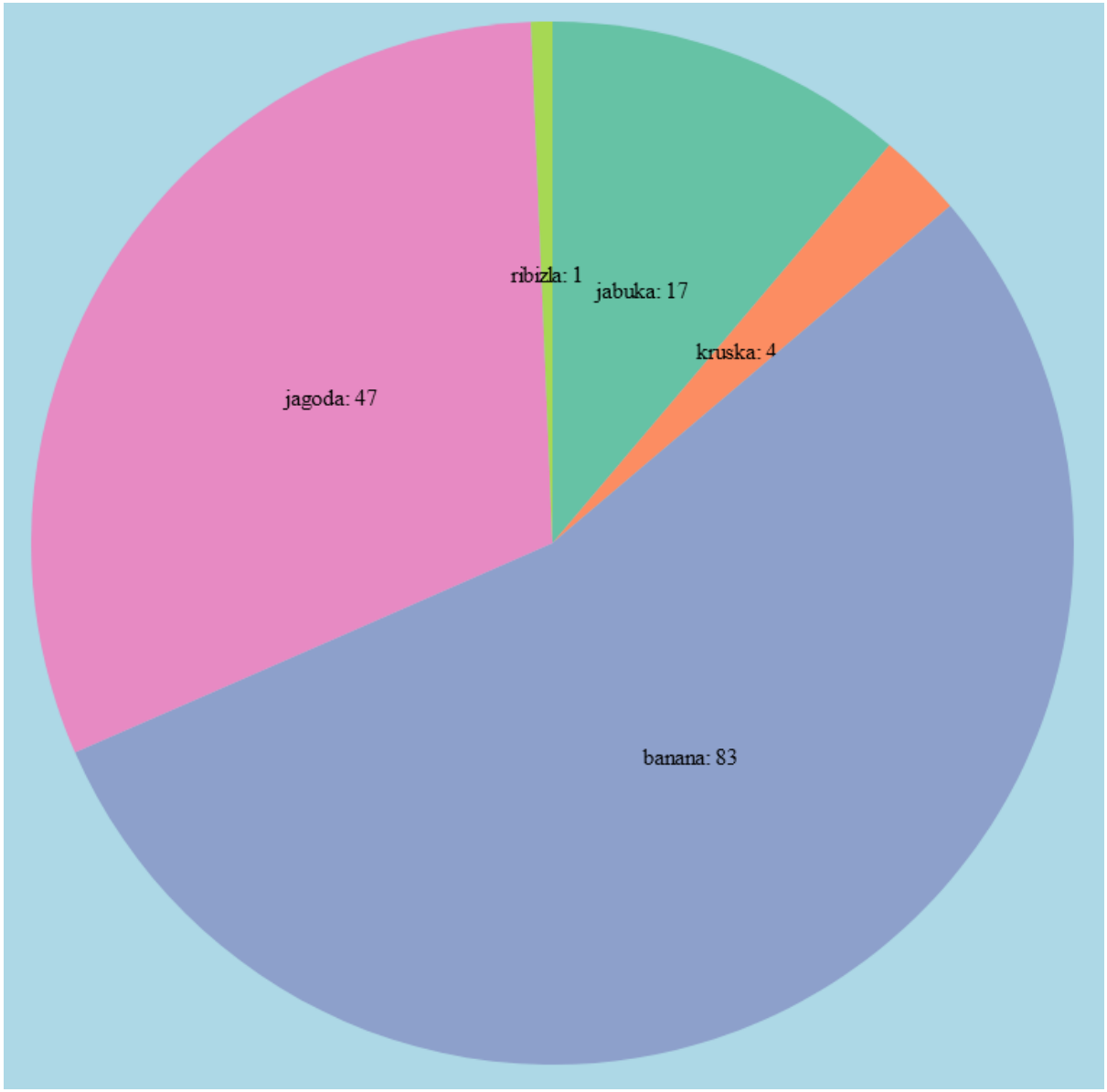
Laboratorijske vježbe iz predmeta Vizualizacija Podataka

Anto Tufeković, FERIT, Osijek, 2020.

Zadatak 1.



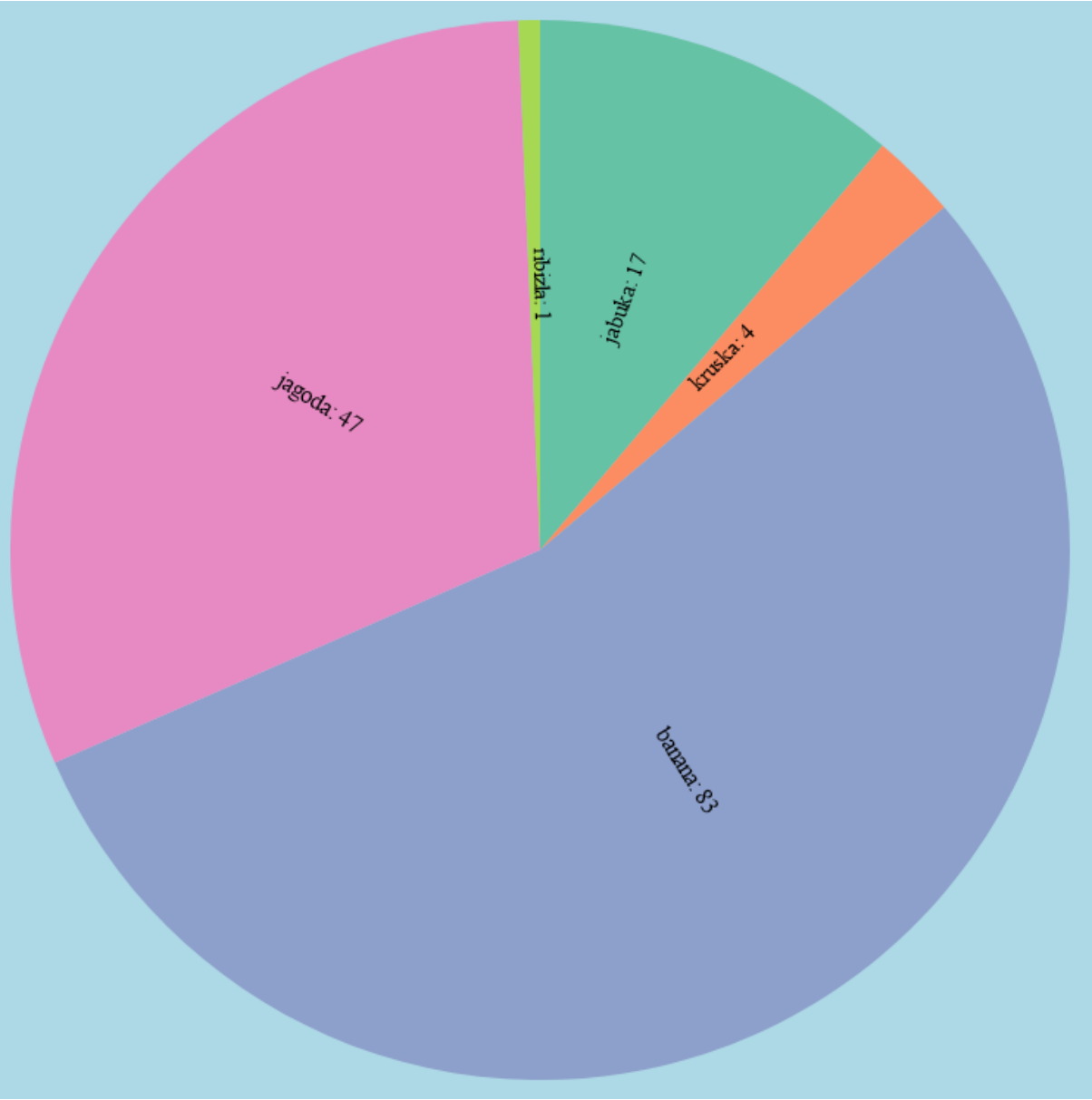
Kod

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Z4-1</title>
    <script src="https://d3js.org/d3.v5.min.js"></script>
</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 = 800;
        var height = 800;
        var outerRadius = Math.min(width, height) / 2;
        outerRadius *= 0.9;
        var innerRadius = 0;
        var svg = d3.select("body")
            .append("svg")
            .attr("width", width)
            .attr("height", height)
            .style("background-color", "lightblue");

        var color = d3.scaleOrdinal(["#66c2a5", "#fc8d62", "#8da0cb", "#e78a
c3", "#a6d854", "#ffd92f"]);
        var pie = d3.pie()
            .value(d => d.value)
            .sort(null);
        var arc = d3.arc()
            .innerRadius(innerRadius)
            .outerRadius(outerRadius);
        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.cent
roid(d) + ")"; })
            .attr("text-anchor", "middle")
            .text(function (d) { return d.data.name + ": " + d.data.value })
;
    </script>
</body>

</html>
```

Kreiramo svg element i prateći predložak kreiramo odgovarajući graf. Jedina razlika je kod kreiranja text elemenata unutar svg elementa vratili smo zajedno sa vrijednošću i ime podatka.

Zadatak 2.



Kod

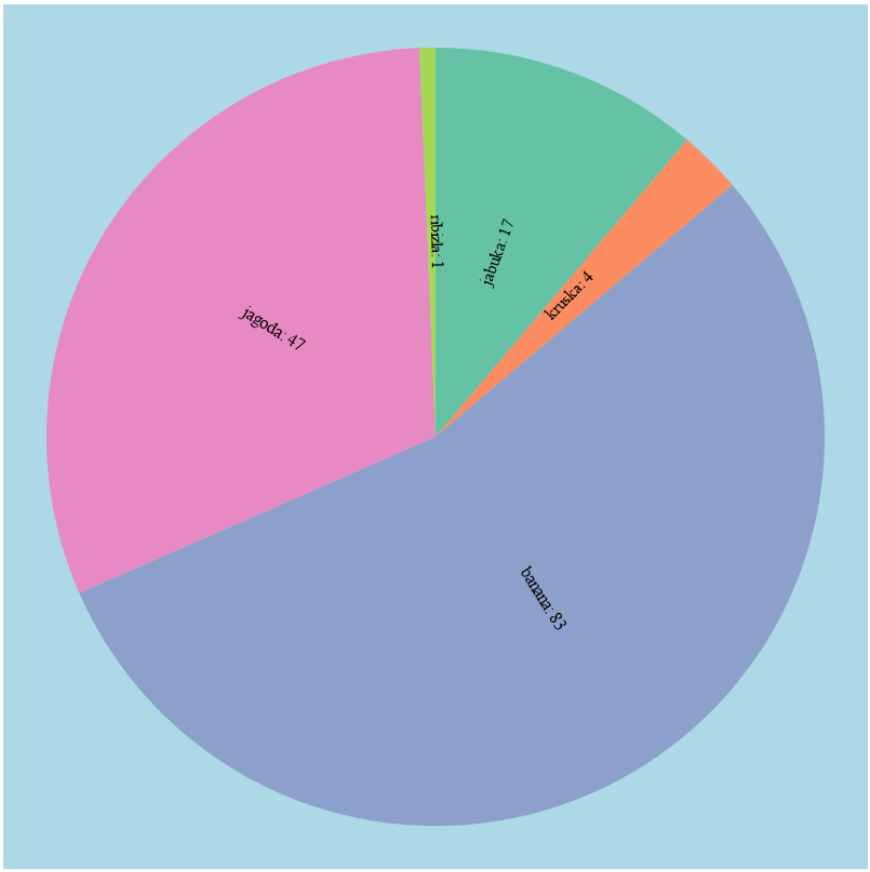
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Z4-2</title>
    <script src="https://d3js.org/d3.v5.min.js"></script>
</head>

<body>
    <script>
        var data = [
```

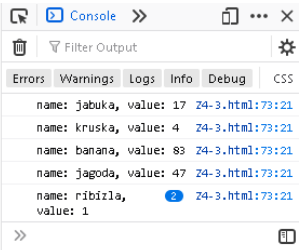
```
            { name: "jabuka", value: 17 },
            { name: "kruska", value: 4 },
            { name: "banana", value: 83 },
            { name: "jagoda", value: 47 },
            { name: "ribizla", value: 1 }
        ];
        var width = 800;
        var height = 800;
        var outerRadius = Math.min(width, height) / 2;
        outerRadius *= 0.9;
        var innerRadius = 0;
        var svg = d3.select("body")
            .append("svg")
            .attr("width", width)
            .attr("height", height)
            .style("background-color", "lightblue");
        var color = d3.scaleOrdinal(["#66c2a5", "#fc8d62", "#8da0cb", "#e78a
c3", "#a6d854", "#ffd92f"]);
        var pie = d3.pie()
            .value(d => d.value)
            .sort(null);
        var arc = d3.arc()
            .innerRadius(innerRadius)
            .outerRadius(outerRadius);

        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);

        var pieArcsText = svg.selectAll("g.text")
            .data(pie(data))
            .enter()
            .append("g")
            .attr("class", "pieText")
            .attr("transform", "translate(" + (width / 2) + ", " + (height /
 2) + ")");
        pieArcsText.append("text")
            .attr("transform", function (d) {
                var sum = 0.0;
                var pi = Math.PI;
                sum += d.endAngle + d.startAngle;
```

```
                sum /= 2;
                sum *= (180 / pi);
                if (sum < 180) {
                    sum -= 90;
                }
                else if (sum >= 180) {
                    sum += 90;
                }
                return "translate(" + arc.centroid(d) + ") rotate(" + sum +
")";
            })
            .attr("text-anchor", "middle")
            .attr("class", "text")
            .text(function (d) { return d.data.name + ": " + d.data.value; }
);

    </script>
</body>

</html>
```

Kod je sličan prvo zadatku, ali pri rješavanju se postupalo drugačije pri kreiranju text elemenata za graf. Prvo se kreirala zasebna grupa za text elementa tako da bi se nactali ispred grupe od grafa. Nakon toga smo pristupili podatcima za njihove početne i krajnje rubove s kojima bi se nacrtao graf. Uzeta je srednja vrijednost i pretvorena je u stupnjeve te tom vrijednošću bi se rotirao text element.

Zadatak 3.





Kod

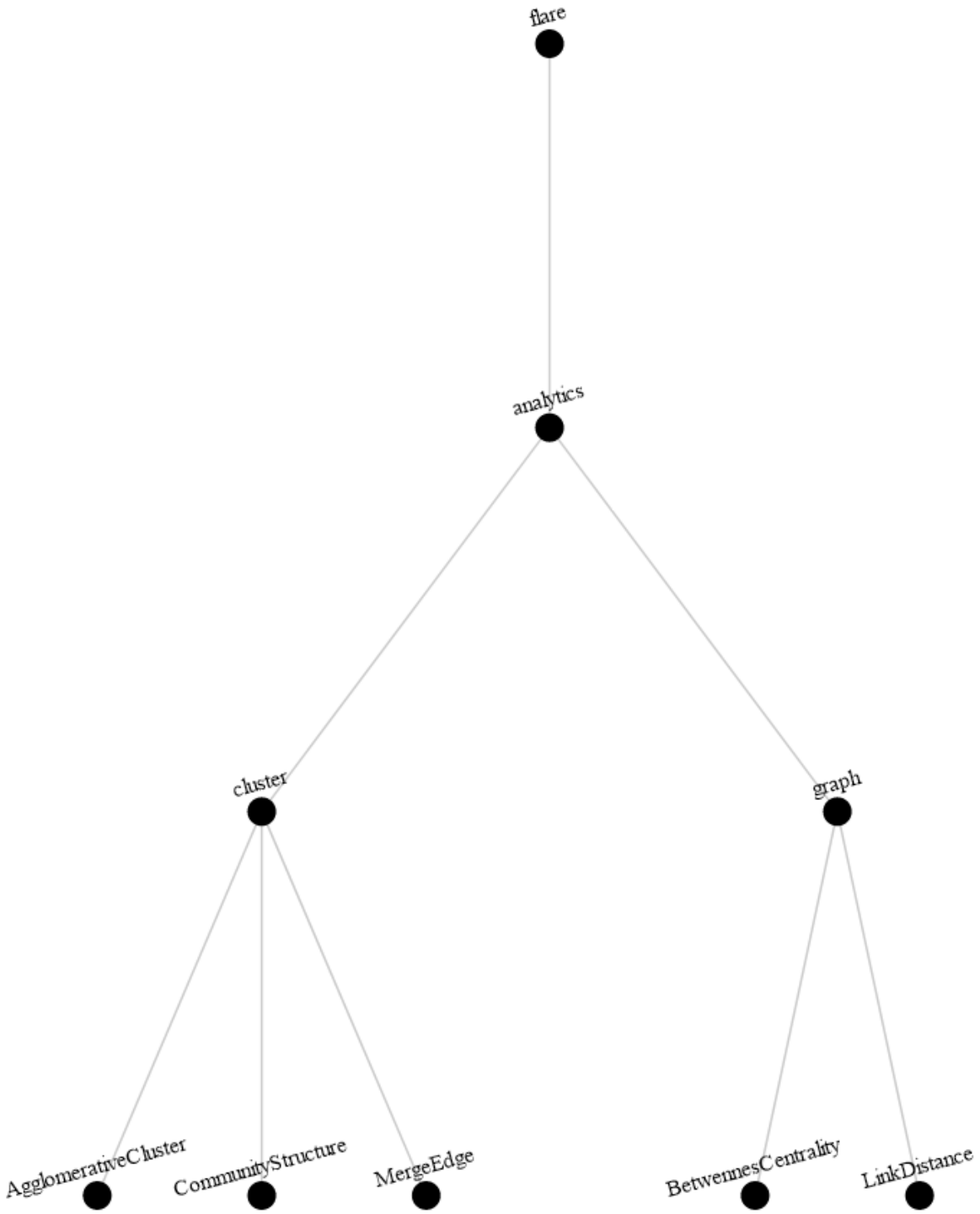
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Z4-3</title>
    <script src="https://d3js.org/d3.v5.min.js"></script>
</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 = 800;
        var height = 800;
```

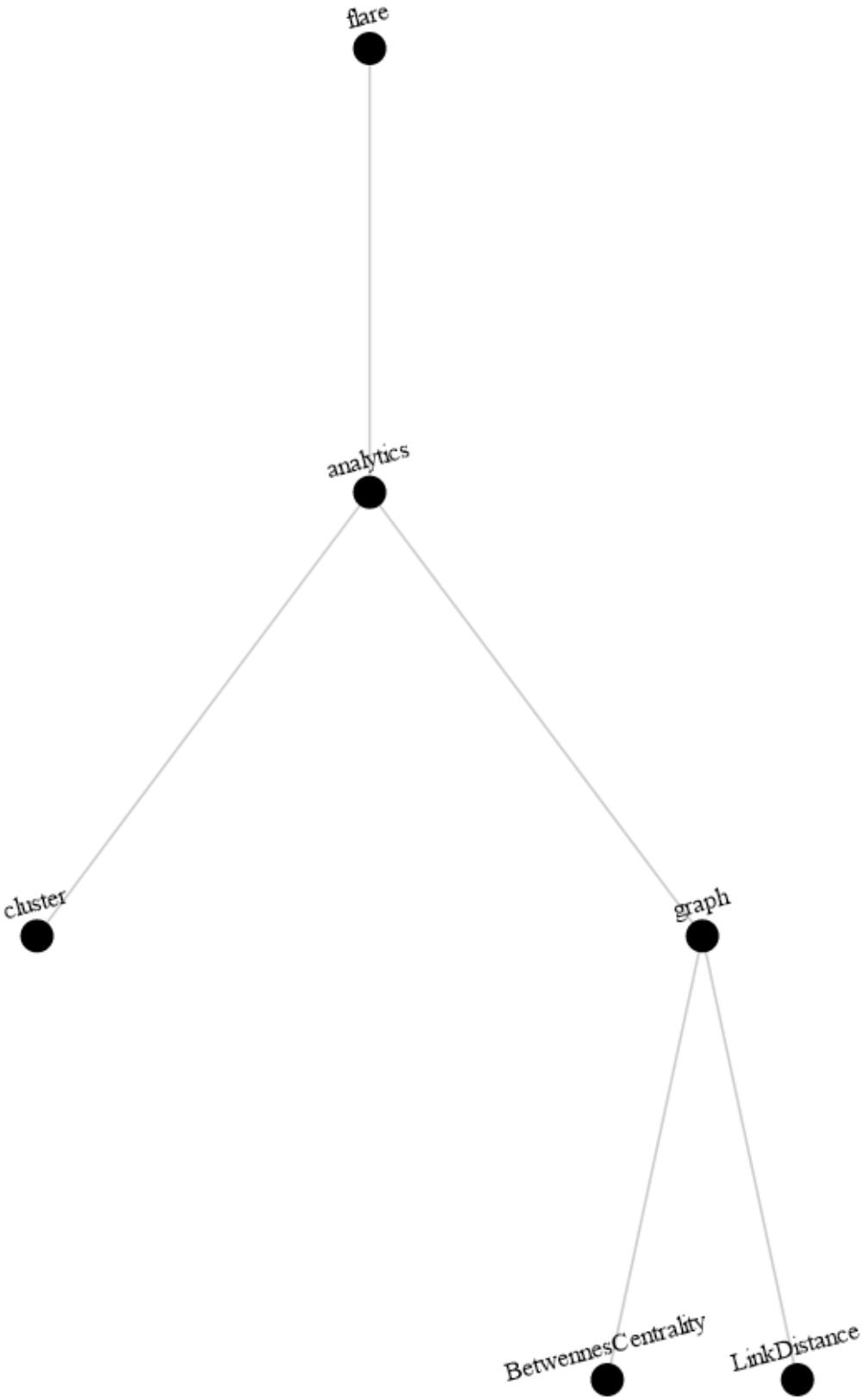
```
        var outerRadius = Math.min(width, height) / 2;
        outerRadius *= 0.9;
        var innerRadius = 0;
        var svg = d3.select("body")
            .append("svg")
            .attr("width", width)
            .attr("height", height)
            .style("background-color", "lightblue");
        var color = d3.scaleOrdinal(["#66c2a5", "#fc8d62", "#8da0cb", "#e78a
c3", "#a6d854", "#ffd92f"]);
        var pie = d3.pie()
            .value(d => d.value)
            .sort(null);
        var arc = d3.arc()
            .innerRadius(innerRadius)
            .outerRadius(outerRadius);

        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);

        var pieArcsText = svg.selectAll("g.text")
            .data(pie(data))
            .enter()
            .append("g")
            .attr("class", "pieText")
            .attr("transform", "translate(" + (width / 2) + ", " + (height /
 2) + ")");
        pieArcsText.append("text")
            .attr("transform", function (d) {
                var sum = 0.0;
                var pi = Math.PI;
                sum += d.endAngle + d.startAngle;
                sum /= 2;
                sum *= (180 / pi);
                if (sum < 180) {
                    sum -= 90;
                }
                else if (sum >= 180) {
                    sum += 90;
                }
```

```
                return "translate(" + arc.centroid(d) + ") rotate(" + sum +
")";
            })
            .attr("text-anchor", "middle")
            .attr("class", "text")
            .text(function (d) { return d.data.name + ": " + d.data.value; }
);
        pieArcs.on("click", function (d) {
            console.log("name: " + d.data.name + ", value: " + d.data.value)
;
        });
    </script>
</body>

</html>
```

Kod je identičan drugom zadatku samo je na kraju dodan onClick promatrač sa anonimnom funkcijom u kojoj se ispisuju u konzoli tražene vrijednosti.

Zadatak 4.


flare
analytics
cluster
graph
AgglomerativeCluster
CommunityStructure
MergeEdge
BetweennesCentrality
LinkDistance



Kod

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Z4-4</title>
    <script src="https://d3js.org/d3.v5.min.js"></script>
    <style>
        .node circle {
            fill: #fff;
            stroke: steelblue;
        }

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

        .link {
            fill: none;
            stroke: #ccc;
            stroke-width: 1.5px;
        }
    </style>
</head>

<body>
    <script>
        var data = {
            "name": "flare",
            "id": 0,
            "childrenID": [1],
            "children": [
                {
                    "name": "analytics",
                    "id": 1,
                    "childrenID": [2, 6],
                    "children": [
                        {
                            "name": "cluster",
                            "id": 2,
                            "childrenID": [3, 4, 5],
                            "children": [
                                { "name": "AgglomerativeCluster", "id": 3, "
childrenID": [], "size": 3938 },
                                { "name": "CommunityStructure", "id": 4, "ch
ildrenID": [], "size": 3812 },
                                { "name": "MergeEdge", "id": 5, "childrenID"
: [], "size": 743 }
                            ]
```

```
                        },
                        {
                            "name": "graph",
                            "id": 6,
                            "childrenID": [7, 8],
                            "children": [
                                { "name": "BetwennesCentrality", "id": 7, "c
hildrenID": [], "size": 3534 },
                                { "name": "LinkDistance", "id": 8, "children
ID": [], "size": 5731 }
                            ]
                        }
                    ]
                }
            ]
        };
        function hideLink(id) {
            link = d3.select("#link-" + id);
            if (link.attr("display") == null || link.attr("display") == "blo
ck") {
                link.attr("display", "none");
            }
            else {
                link.attr("display", "block");
            }
        }
        function hideNode(id) {
            node = d3.select("#node-" + id);
            if (node.attr("display") == null || node.attr("display") == "blo
ck") {
                node.attr("display", "none");
            }
            else {
                node.attr("display", "block");
            }
        }
        function hideText(id) {
            text = d3.select("#text-" + id);
            if (text.attr("display") == null || text.attr("display") == "blo
ck") {
                text.attr("display", "none")
            }
            else {
                text.attr("display", "block");
            }
        }


        function hideDescendants(d) {
```

```
        if (d.data.childrenID.length > 0) {
            d.data.childrenID.forEach(element => {
                console.log("elementID:" + element);
                var next;
                d.data.children.forEach(child => {
                    if (child.id == element) {
                        next = child;
                        console.log("Chosen next:");
                        console.log(next);
                        hideDescendantsAndSelf(element, next);
                    }
                    else {
                        next = "Not found";
                    }
                });
            });
        }
    };
    function hideDescendantsAndSelf(id, data) {
        hideText(id);
        hideNode(id);
        hideLink(id);
        if (data.childrenID.length > 0) {
            data.childrenID.forEach(element => {
                console.log("elementID:" + element);
                var next;
                data.children.forEach(child => {
                    if (child.id == element) {
                        next = child;
                        console.log("Chosen next:");
                        console.log(next);
                        hideDescendantsAndSelf(element, next);
                    }
                    else {
                        next = "Not found"
                    }
                })
            })
        }
    };

    var width = 900;
    var height = 900;

    var cluster = d3.cluster()
        .size([height - 100, width - 100]);

    var svg = d3.select("body")
```

```
            .append("svg")
            .attr("width", width)
            .attr("height", height);

        var container = svg.append("g")
            .attr("transform", "translate(40, 40)");

        var root = d3.hierarchy(data);
        cluster(root);

        links = container.selectAll("line.link")
            .data(root.links())
            .enter()
            .append("line")
            .classed("link", true)
            .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; })
            .attr("id", function (d, i) { return "link-
" + d.target.data.id; });

        nodes = container.selectAll("circle.nodes")
            .data(root.descendants())
            .enter()
            .append("circle")
            .classed("node", true)
            .attr("id", function (d) { return "node-" + d.data.id; })
            .attr("cx", function (d) { return d.x; })
            .attr("cy", function (d) { return d.y; })
            .attr("r", 10)
            .on("click", function (d) {
                hideDescendants(d);
            });

        texts = container.selectAll("text")
            .data(root.descendants())
            .enter()
            .append("text")
            .attr("transform", function (d) { return "translate(" + (d.x) +
", " + (d.y - 15) + ") rotate(-15)"; })
            .attr("text-anchor", "middle")
            .attr("id", function (d) { return "text-" + d.data.id; })
            .text(function (d) { return d.data.name; });
    </script>
</body>

</html>
```

Dosad najteži zadatak, pratiti predložak nije bilo moguće zbog razlike verzija d3 biblioteke (predložak koristi v3 dok je trenutna verzija v5). Nakon što sam skontao kako napraviti pravilno graf, odlučio sam preko id-eva napraviti rekurzivno sakrivanje djece nakon klika. U funkciji se pravi par provjera dali je pravilno djete odabrano te se onda preko tog djeteta dalje propagira funkcija. Ako nema djece onda se samo sakrije/otkrije bez da išta dalje radi. Prilikom sakrivanja/otkrivanja koristi se činjenica da postoje unikatni identifikatori za svaki podatak (za njegovu liniju od roditelja, za njegovu točku u svg-u i njegov tekst, sve troje se onda sakriva).