Kauno technologijos universitetas

Multimedijos inžinerijos katedra

Trapecijų žemėlapio sudarymo sudarymo įgyvendimas JavaScript kalboje

Skaičuojamosios geometrijos algoritmai ir duomenų struktūros (P170M106) kursinio darbo ataskaita

|  |  |
| --- | --- |
| **Atliko:** | IFM-6/1 Povilas Simanatis |
|  | (Grupė, vardas, pavardė, parašas) |
|  |  |
| **Dėstytojas:** | *Doc. Dr. Gintaras Palubeckis* |
|  | (Pareigos, laipsnis, vardas, pavardė) |
|  | (Įvertinimas, parašas) |

**Kaunas, 2016**

**Turinys**

[1 Užduotis 3](#_Toc469651970)

[1.1 Pseudo kodas 3](#_Toc469651971)

[1.2 Programos kodas 3](#_Toc469651972)

[1.3 Rezultatai 9](#_Toc469651973)

[1.4 Išvados 12](#_Toc469651974)

# Užduotis

Suskaidyti duotąją segmentų aibę trapecijomis sudarant trapecijų žemėlapį.

## Pseudo kodas

Progamos pseudo kodas

Segmentai = SugeneruokAtsiktiktiniusSegmentus()

TrapecijųLinijos = PapildykTrapecijosLinijomis()

Atvaizduok(Segmentai);

Atvaizduok(TrapecijųLinijas)

PapildykTrapecijosLinijomis(){

SupančiosTrapeijos = GautSupančiasTrapecijas();

Jei(SupančiosTrapecijos.yraViršuj){

TrapecijosLinijosViršus = AtstumasIki (SupančiosTrapecijos.viršus);

}

Jei(SupančiosTrapecijos.yraApačioje){

TrapecijosLinijosApačia = AtstumasIkiSupančiosTrapecijos(SupančiosTrapecijos.apačia);

}

Gražink TrapecijosLinija

}

## Programos kodas

var c = document.getElementById('myCanvas');

var ctx = c.getContext('2d');

ctx.canvas.width = 400;

ctx.canvas.height = 400;

function spausdinkDuomenis(selector, data) {

var element = document.querySelectorAll(selector);

var formedData = `

<table class="table table-bordered table-striped table-condensed">

<tr>

<th>Segmento Nr</th>

<th>Pradžios taškas</th>

<th>Pabaigos taškas</th>

</tr>

`;

data.forEach(function (segment, index) {

formedData += `

<tr>

<td>

${index + 1}

</td>

<td>

${JSON.stringify(segment.p)}

</td>

<td>

${JSON.stringify(segment.q)}

</td>

</tr>

` });

formedData += `</table>`;

element[0].innerHTML = formedData;

};

function roundFloat(number, precision) {

var factor = Math.pow(10, precision);

var tempNumber = number \* factor;

var roundedTempNumber = Math.round(tempNumber);

return roundedTempNumber / factor;

};

function drawLine(segmentas, spalva) {

ctx.beginPath();

ctx.moveTo(segmentas.p.x, segmentas.p.y);

ctx.lineTo(segmentas.q.x, segmentas.q.y);

ctx.lineWidth = 0.5;

ctx.strokeStyle = spalva;

ctx.stroke();

}

function vienodas(segmentas, kitas) {

return (

segmentas.p.x == kitas.p.x

&& segmentas.p.y == kitas.p.y

&& segmentas.q.x == kitas.q.x

&& segmentas.q.y == kitas.q.y

)

};

function vertikalusAtstumasTS(taskas, segmentas) {

if (taskas.x == segmentas.p.x) return segmentas.p.y - taskas.y;

if (taskas.x == segmentas.q.x) return segmentas.q.y - taskas.y;

if (taskas.x > segmentas.p.x && taskas.x < segmentas.q.x) {

var ilgis = segmentas.p.x - segmentas.q.x

var aukstis = segmentas.p.y - segmentas.q.y;

var kitasAukstis = segmentas.p.y - taskas.y;

if (!ilgis || !aukstis) return taskas.y - segmentas.p.y;

var aIlgis = segmentas.p.x - taskas.x;

//turetu buti return kitasAukstis + (aIlgis \* aukstis / ilgis);

return roundFloat(aukstis + (aIlgis \* aukstis / ilgis), 2);

}

return false

};

function deltaH(taskas, segmentas) {

if (taskas.x == segmentas.p.x) return segmentas.p.y - taskas.y;

if (taskas.x == segmentas.q.x) return segmentas.q.y - taskas.y;

if (taskas.x > segmentas.p.x && taskas.x < segmentas.q.x) {

var ilgis = segmentas.p.x - segmentas.q.x

var aukstis = segmentas.p.y - segmentas.q.y;

if (!ilgis || !aukstis) return taskas.y - segmentas.p.y;

var aIlgis = segmentas.p.x - taskas.x;

return roundFloat((aIlgis \* aukstis / ilgis), 2);

}

return false

};

function generuokSegmentus(segmentai, count) {

for (var i = 0; i < count; i++) {

segmentai.push(

{

p: { x: getRandomInt(10, 400), y: getRandomInt(10, 400) },

q: { x: getRandomInt(0, 400), y: getRandomInt(0, 400) }

}

);

}

return segmentai;

}

function atvaizduok(segmentai, spalva) {

segmentai.forEach(function (segmentas) {

drawLine(segmentas, spalva);

});

}

function getRandomInt(min, max) {

return Math.floor(Math.random() \* (max - min + 1)) + min;

}

var pradiniaiSegmentai = [

{

p: { x: 10, y: 50 },

q: { x: 190, y: 10 }

},

{

p: { x: 45, y: 100 },

q: { x: 180, y: 100 }

},

{

p: { x: 10, y: 250 },

q: { x: 190, y: 200 }

},

{

p: { x: 10, y: 150 },

q: { x: 190, y: 165 }

},

{

p: { x: 5, y: 200 },

q: { x: 195, y: 230 }

},

{

p: { x: 40, y: 50 },

q: { x: 380, y: 380 }

}

];

var segmentai = generuokSegmentus(pradiniaiSegmentai, 0);

console.log(JSON.stringify(segmentai));

var trapecijuLinijos = papikdykTrapecijuLinijom(segmentai, "red");

atvaizduok(segmentai, "black");

atvaizduok(trapecijuLinijos, "red");

console.log(JSON.stringify(trapecijuLinijos));

spausdinkDuomenis('#segments', segmentai);

spausdinkDuomenis('#TLines', trapecijuLinijos);

function papikdykTrapecijuLinijom(segmentai, spalva, p) {

function aplinkSegmentai(segmentas, trapecijosLinija, p) {

var gretimi = {

top: [],

topAtstumas: 400,

bottom: [],

bottomAtstumas: -400

};

segmentai.forEach(function (kitas) {

if (p) {

var atstumas = vertikalusAtstumasTS(segmentas.p, kitas);

} else {

var atstumas = vertikalusAtstumasTS(segmentas.q, kitas);

}

if (!vienodas(segmentas, kitas)) {

if (atstumas && (segmentas.p.y >= kitas.p.y)) {

if (Math.abs(atstumas) < Math.abs(gretimi.topAtstumas)) {

gretimi.topAtstumas = atstumas;

gretimi.top.pop();

gretimi.top.push(kitas);

}

}

if (atstumas && (segmentas.p.y <= kitas.p.y)) {

if (Math.abs(atstumas) < Math.abs(gretimi.bottomAtstumas)) {

gretimi.bottomAtstumas = atstumas;

gretimi.bottom.pop();

gretimi.bottom.push(kitas);

}

}

}

});

return gretimi;

}

var trapecijuLinijos = segmentai.map(function (segmentas, index) {

var tempLinija = {

p: {

x: segmentas.p.x,

y: 0

},

q: {

x: segmentas.p.x,

y: 400

}

};

var gretimiSegmentai = aplinkSegmentai(segmentas, tempLinija, true);

if (index === 6) {

console.log(segmentas, gretimiSegmentai);

}

if (gretimiSegmentai.top[0]) {

tempLinija.p.y = gretimiSegmentai.top[0].p.y;

if ((gretimiSegmentai.top[0].p.y != gretimiSegmentai.top[0].q.y) && (gretimiSegmentai.top[0].p.x != segmentas.p.x)) {

tempLinija.p.y -= deltaH(

{

x: segmentas.p.x,

y: gretimiSegmentai.top[0].p.y

},

gretimiSegmentai.top[0]

);

}

}

if (gretimiSegmentai.bottom[0]) {

tempLinija.q.y = gretimiSegmentai.bottom[0].p.y;

if ((gretimiSegmentai.bottom[0].p.y != gretimiSegmentai.bottom[0].q.y) && (gretimiSegmentai.bottom[0].p.x != segmentas.p.x)) {

tempLinija.q.y -= deltaH(

{

x: segmentas.p.x,

y: gretimiSegmentai.bottom[0].p.y

},

gretimiSegmentai.bottom[0]

);

}

};

return tempLinija;

});

trapecijuLinijos = trapecijuLinijos.concat(segmentai.map(function (segmentas) {

var tempLinija = {

p: {

x: segmentas.q.x,

y: 0

},

q: {

x: segmentas.q.x,

y: 400

}

};

var gretimiSegmentai = aplinkSegmentai(segmentas, tempLinija, false);

if (gretimiSegmentai.top[0]) {

tempLinija.p.y = gretimiSegmentai.top[0].p.y;

if ((gretimiSegmentai.top[0].p.y != gretimiSegmentai.top[0].q.y) && (gretimiSegmentai.top[0].q.x != segmentas.q.x)) {

tempLinija.p.y -= deltaH(

{

x: segmentas.q.x,

y: gretimiSegmentai.top[0].p.y

},

gretimiSegmentai.top[0]

);

}

}

if (gretimiSegmentai.bottom[0]) {

tempLinija.q.y = gretimiSegmentai.bottom[0].p.y;

if ((gretimiSegmentai.bottom[0].p.y != gretimiSegmentai.bottom[0].q.y) && (gretimiSegmentai.bottom[0].q.x != segmentas.q.x)) {

tempLinija.q.y -= deltaH(

{

x: segmentas.q.x,

y: gretimiSegmentai.bottom[0].q.y

},

gretimiSegmentai.bottom[0]

);

}

};

return tempLinija;

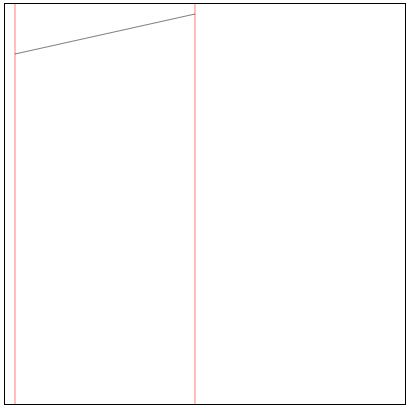
}));

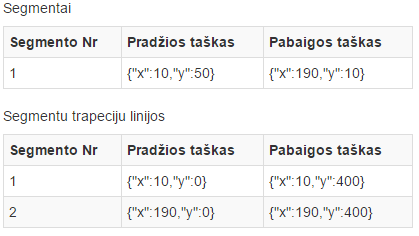
return trapecijuLinijos;

}

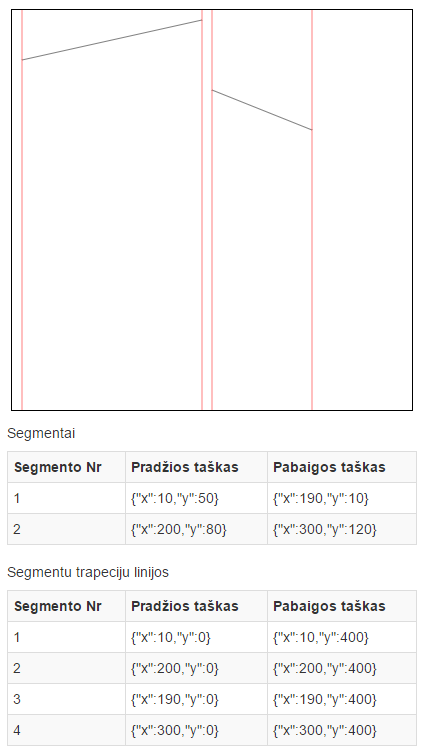
## Rezultatai

Vieno segmento trapecijų žemėlapis

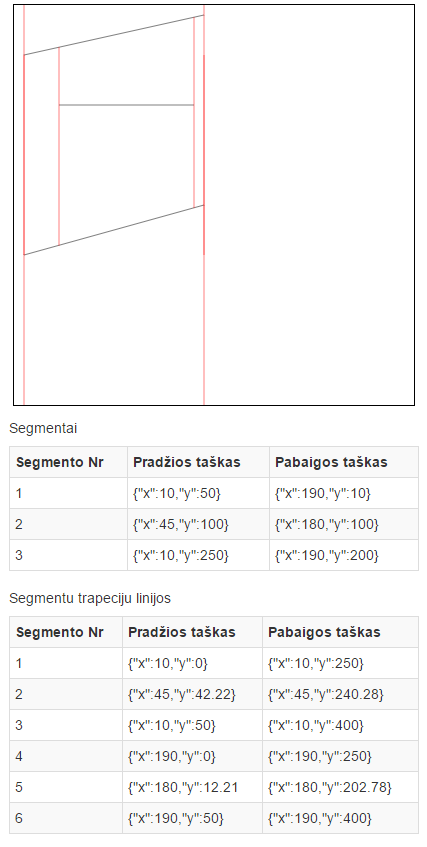




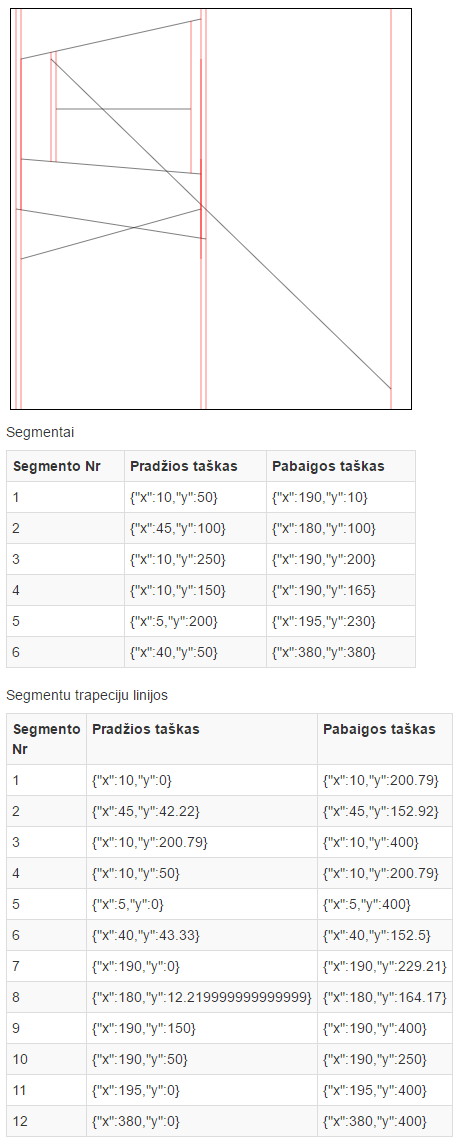
Dviejų nelygiagrečių segmentų žemėlapis



Keleto lygiagrečių segmentų žemėlapis



Segmentų žemėlapis papildytas atsitiktiniais duomenimis



## Išvados

Trapecijų žemėlapio generavimas, nors ir atrodo paprasta užduotis, tačiau ją įgyvendinti optimizuojant trapecijų linijas buvo sudėtinga, kadangi artimiausių kaimynų nustatymą atstumų, bei kitus skaičiavimus teko rašyti nuo pradžių. Kita vertus Javascript kalbos pasirinkimas suteikia programai skaitomumo ir paprastumo, nes nereikia rūpintis kintamųjų ir funkcijų tipais, kaip reiktų daryti tipizuotose kalbose.