

**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №4**  
з дисципліни  
«Алгоритми і структури даних»

Виконав:

Студент групи ІМ-41  
Добровольський Антон Володимирович  
номер у списку групи: 8

Перевірив:

Сергієнко А.М.

Київ 2024

## Задача:

Потрібно зобразити і обчислити різноманітні операції про граfi

Мій сід був 4108. На жаль, у JavaScript немає вбудованого генератора сідом, тому я зробив простий генератор

## Текст програми:

JS

```
const canvasD = document.getElementById('canvasDirected');
const canvasU = document.getElementById('canvasUndirected');
const canvasM = document.getElementById('canvasModified');
const canvasC = document.getElementById('canvasCondensed');
const n1 = 4;
const n2 = 1;
const n3 = 0;
const n4 = 8;
const k1 = 1 - (n3 * 0.01) - (n4 * 0.01) - 0.3;
const k2 = 1 - (n3 * 0.005) - (n4 * 0.005) - 0.27;
const nodeNumber = 10 + n3;
const nodeRadius = 15;
const nodes = [
  {x: 100, y: 100},
  {x: 300, y: 100},
  {x: 500, y: 100},
  {x: 700, y: 100},
  {x: 100, y: 300},
  {x: 400, y: 300},
  {x: 700, y: 300},
  {x: 100, y: 500},
  {x: 400, y: 500},
  {x: 700, y: 500},
]
const arraysEqual = (a, b) => {
  if (a.length !== b.length) return false;
  return a.every((val, i) => val === b[i]);
}
const randomizeArray = () => {
const result = [];
for (let i = 0; i < nodeNumber; i++) {
  const row = [];
  for (let j = 0; j < nodeNumber; j++) {
    row.push(Math.random() * 2);
  }
  result.push(row);
} return result };
const rArray = randomizeArray();
const graphifyArray = (array, k) => {
  const result = [];
  for (let i = 0; i < nodeNumber; i++) {
    const row = [];
    for (let j = 0; j < nodeNumber; j++) {
      const value = array[i][j];
      const newValue = Math.floor(value * k);
      row.push(newValue);
    } result.push(row);
  }
```

```

    } return result;
}
const undirectArray = (array) => {
    const result = [];
    for (let i = 0; i < nodeNumber; i++) {
        const row = [];
        for (let j = 0; j < nodeNumber; j++) {
            row.push(Math.max(array[i][j], array[j][i]))
        } result.push(row);
    } return result;
}

const dArray = graphifyArray(rArray, k1);
const uArray = undirectArray(dArray);
const mArray = graphifyArray(rArray, k2);
console.log('Напрямлений граф:');
console.log(dArray);
console.log('Ненапрямлений граф:');
console.log(uArray);
console.log('Модифікований граф:');
console.log(mArray);

const drawCurve = (ctx, x1, y1, x2, y2, arrowed) => {
    // Коригуємо точки з урахуванням радіусу вершин
    const angle = Math.atan2(y2 - y1, x2 - x1);
    const adjustedX1 = x1 + 15 * Math.cos(angle);
    const adjustedY1 = y1 + 15 * Math.sin(angle);
    const adjustedX2 = x2 - 15 * Math.cos(angle);
    const adjustedY2 = y2 - 15 * Math.sin(angle);
    const midX = (adjustedX1 + adjustedX2) / 2;
    const midY = (adjustedY1 + adjustedY2) / 2; // Медіана
    const dx = Math.abs(adjustedX2 - adjustedX1);
    const dy = Math.abs(adjustedY2 - adjustedY1);
    const len = Math.sqrt(dx*dx + dy*dy);
    const nx = -dy/len;
    const ny = dx/len; // Вектор перпендикуляра до лінії
    const offset = len * 0.2;
    const controlX = midX + nx * offset;
    const controlY = midY + ny * offset; // Зміщення для кривизни
    ctx.beginPath();
    ctx.moveTo(adjustedX1, adjustedY1);
    ctx.quadraticCurveTo(controlX, controlY, adjustedX2, adjustedY2);
    ctx.strokeStyle = '#666';
    ctx.lineWidth = 2;
    ctx.stroke();

    if (arrowed) {
        const arrowAngle = Math.atan2(adjustedY2 - controlY, adjustedX2 - controlX);
        ctx.beginPath();
        ctx.moveTo(adjustedX2, adjustedY2);
        ctx.lineTo(
            adjustedX2 - 10 * Math.cos(arrowAngle - Math.PI/6),
            adjustedY2 - 10 * Math.sin(arrowAngle - Math.PI/6)
        );
        ctx.lineTo(
            adjustedX2 - 10 * Math.cos(arrowAngle + Math.PI/6),
            adjustedY2 - 10 * Math.sin(arrowAngle + Math.PI/6)
        );
    }
};

```

```

    ctx.closePath();
    ctx.fillStyle = '#666';
    ctx.fill();}
};
const drawLoop = (ctx, x, y) => {
    ctx.beginPath();
    ctx.arc(x, y - 15, 25, Math.PI/2, Math.PI*2.5);
    ctx.strokeStyle = '#666';
    ctx.lineWidth = 2;
    ctx.stroke();
};
const drawGraph = (canvas, nodes, matrix, directed) => {
    const ctx = canvas.getContext('2d');
    matrix.forEach((row, i) => {
        row.forEach((value, j) => {
            if (value === 1) {
                const node1 = nodes[i];
                const node2 = nodes[j];
                if (i === j) {drawLoop(ctx, node1.x, node1.y)}
                else {drawCurve(ctx, node1.x, node1.y, node2.x, node2.y, directed)}
            }
        });
    });
    nodes.forEach((node, id) => {
        ctx.beginPath();
        ctx.arc(node.x, node.y, 15, 0, Math.PI * 2);
        ctx.fillStyle = '#f0f0f0';
        ctx.fill();
        ctx.fillStyle = '#000';
        ctx.font = '14px Arial';
        ctx.textAlign = 'center';
        ctx.textBaseline = 'middle';
        ctx.fillText(id, node.x, node.y);
    });
}

```

```

drawGraph(canvasD, nodes, dArray, true);
drawGraph(canvasU, nodes, uArray, false);
drawGraph(canvasM, nodes, mArray, true);

```

```

const secondStep = (darray, uarray) => {
    console.log('Крок 2:');
    console.log('Степені вершин ненапрявленого графа:');
    const degrees = [];
    let regularGraph = true;
    for (let i = 0; i < nodeNumber; i++) {
        let sum = uarray[i][i];
        for (let j = 0; j < nodeNumber; j++) {
            sum += uarray[i][j];
        } console.log('Степень вершини ' + i + ': ' + sum);
        if (uarray[i][i] === 1) {sum -= 2}
        degrees.push(sum);
    }
    console.log('Степені вершин направленного графа:');
    for (let i = 0; i < nodeNumber; i++) {
        let degree = {вхід: 0, вихід: 0, всього: 0};
        for (let j = 0; j < nodeNumber; j++) {

```

```

        if (darray[i][j] === 1) degree.вихід++
        if (darray[j][i] === 1) degree.вхід++
    }
    degree.всього = degree.вхід + degree.вихід;
    console.log('Степінь вершини ' + i + ": " + degree.вхід + '(вхід) ' +
degree.вихід + '(вихід) ' + degree.всього + '(всього)');
}
degrees.forEach((degree, i) => {
    if (degree !== degrees[0]) regularGraph = false;
    if (degree === 0) console.log('Вершина ' + i + ' ізольована');
    if (degree === 1) console.log('Вершина ' + i + ' висяча');
});
if (regularGraph) console.log('Граф є регулярним зі степенем' + degrees[0]);
else console.log('Граф не є регулярним');
});
secondStep(dArray, uArray);

const fourthStep = (array) => {
    console.log('Степені вершин модифікованого графа:');
    for (let i = 0; i < nodeNumber; i++) {
        let degree = {вхід: 0, вихід: 0};
        for (let j = 0; j < nodeNumber; j++) {
            if (array[i][j] === 1) degree.вихід++
            if (array[j][i] === 1) degree.вхід++
        }
        console.log('Степінь вершини ' + i + ": " + degree.вхід + '(вхід) ' +
degree.вихід + '(вихід) ');
    }
    let connections = new Set();
    let twoWays = new Set();
    let threeWays = new Set();
    for (let i = 0; i < nodeNumber; i++) {
        for (let j = 0; j < nodeNumber; j++) {
            if (array[i][j] === 1) connections.add(JSON.stringify([i, j]));
        }
    }
    console.log("Усі шляхи довжини 2 і 3:")
    connections.forEach((cjson) => {
        const c = JSON.parse(cjson);
        for (let i = 0; i < nodeNumber; i++) {
            if (array[c[1]][i] === 1) { twoWays.add(JSON.stringify([c[0], c[1], i]))}
        }
    });
    twoWays.forEach((twjson) => {
        console.log(twjson);
        const tw = JSON.parse(twjson);
        for (let i = 0; i < nodeNumber; i++) {
            if (array[tw[2]][i] === 1) { threeWays.add(JSON.stringify([tw[0], tw[1],
tw[2], i]))}
        }
    });
    threeWays.forEach((trjson) => {console.log(trjson)});
    let reachMatrix = JSON.parse(JSON.stringify(array));
    let rMc = true;
    console.log("Матриця досяжності:");
    while (rMc) {
        rMc = false;
        for (let i = 0; i < nodeNumber; i++) {
            for (let j = 0; j < nodeNumber; j++) {
                if (reachMatrix[i][j] === 1) {
                    for (let q = 0; q < nodeNumber; q++) {

```

```

        if (reachMatrix[j][q] === 1 && reachMatrix[i][q] === 0) {
            reachMatrix[i][q] = 1;
            rMc = true}}}}});
console.log(reachMatrix);
console.log("Матриця сильної зв'язності:");
const conMatrix = [];
for (let i = 0; i < nodeNumber; i++) {
    const row = [];
    for (let j = 0; j < nodeNumber; j++) {
        if (reachMatrix[i][j] === reachMatrix[j][i] && reachMatrix[i][j] === 1)
row.push(1);
        else row.push(0);
    }
    conMatrix.push(row);
};
console.log(conMatrix);
console.log("Перелік КСЗ:");
let groups = [];
const visits = new Array(nodeNumber).fill(false);
for (let i = 0; i < nodeNumber; i++) {
    if (!visits[i]) {
        const group = [];
        for (let j = 0; j < nodeNumber; j++) {
            if (conMatrix[i][j] === 1 || j === i) {
                group.push(j);
                visits[j] = true}};
        groups.push(group)}};
console.log(groups);
const nodeCNumber = groups.length;
const nodesC = new Array(nodeCNumber);
const matrixC = [];
for (let i = 0; i < nodeCNumber; i++) {
    nodesC[i] = nodes[i];
    const rowC = [];
    for (let j = 0; j < nodeCNumber; j++) {
        if (i === j) {rowC.push(0)}
        else {
            let whatPush = 0;
            const groupi = groups[i];
            const groupj = groups[j];
            groupi.forEach((nodei) => {
                groupj.forEach((nodej) => {
                    if (array[nodei][nodej] === 1) {whatPush = 1}})})
            rowC.push(whatPush);
        }
    }
    matrixC.push(rowC);
};
drawGraph(canvasC, nodesC, matrixC, true);
}
fourthStep(mArray);

```

## HTML

```

<!DOCTYPE html>
<html>
<head>
    <title>Лабораторна 4</title>
    <style>
        canvas { border: 1px solid #000000; }
    </style>

```

```

</style>
</head>
<body>
  <canvas id="canvasDirected" width="800" height="600"></canvas>
  <canvas id="canvasUndirected" width="800" height="600"></canvas>
  <canvas id="canvasModified" width="800" height="600"></canvas>
  <canvas id="canvasCondensed" width="800" height="600"></canvas>
  <script src="main.js"></script>
</body>
</html>

```

## Тестування програми:

### Пункт 1:

Напрямлений граф:

```

(10) [Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10)]
  0: (10) [0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
  1: (10) [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  2: (10) [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
  3: (10) [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
  4: (10) [0, 0, 1, 0, 0, 0, 0, 1, 0, 0]
  5: (10) [0, 0, 1, 0, 0, 0, 1, 0, 0, 0]
  6: (10) [0, 0, 0, 1, 0, 0, 1, 0, 0, 0]
  7: (10) [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  8: (10) [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
  9: (10) [0, 0, 1, 0, 1, 0, 0, 0, 1, 1]
    length: 10
    [[Prototype]]: Array(0)

```

Ненапрямлений граф:

```

(10) [Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10)]
  0: (10) [0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
  1: (10) [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
  2: (10) [0, 1, 0, 0, 1, 1, 0, 0, 0, 1]
  3: (10) [1, 0, 0, 1, 0, 0, 1, 0, 0, 0]
  4: (10) [0, 0, 1, 0, 0, 0, 0, 1, 0, 1]
  5: (10) [0, 0, 1, 0, 0, 0, 1, 0, 0, 0]
  6: (10) [0, 0, 0, 1, 0, 1, 1, 0, 1, 0]
  7: (10) [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
  8: (10) [1, 0, 0, 0, 0, 0, 1, 0, 0, 1]
  9: (10) [0, 0, 1, 0, 1, 0, 0, 0, 1, 1]
    length: 10

```

### Пункт 2:

Крок 2:

Степені вершин ненапряженого графа:

Степінь вершини 0: 2

Степінь вершини 1: 1

Степінь вершини 2: 4

Степінь вершини 3: 4

Степінь вершини 4: 3

Степінь вершини 5: 2

Степінь вершини 6: 5

Степінь вершини 7: 1

Степінь вершини 8: 3

Степінь вершини 9: 5

Степені вершин направленного графа:

Степінь вершини 0: 0(вхід) 2(вихід) 2(всього)

Степінь вершини 1: 1(вхід) 0(вихід) 1(всього)

Степінь вершини 2: 3(вхід) 1(вихід) 4(всього)

Степінь вершини 3: 3(вхід) 1(вихід) 4(всього)

Степінь вершини 4: 1(вхід) 2(вихід) 3(всього)

Степінь вершини 5: 0(вхід) 2(вихід) 2(всього)

Степінь вершини 6: 3(вхід) 2(вихід) 5(всього)

Степінь вершини 7: 1(вхід) 0(вихід) 1(всього)

Степінь вершини 8: 2(вхід) 1(вихід) 3(всього)

Степінь вершини 9: 1(вхід) 4(вихід) 5(всього)

Вершина 1 висяча

Вершина 7 висяча

Граф не є регулярним

## Пункт 3:

Модифікований граф:

```
(10) [Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10)]
  0: (10) [0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
  1: (10) [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
  2: (10) [0, 1, 1, 0, 0, 0, 1, 0, 0, 0]
  3: (10) [0, 0, 0, 1, 1, 0, 1, 0, 0, 0]
  4: (10) [0, 0, 1, 0, 0, 0, 0, 1, 0, 0]
  5: (10) [0, 0, 1, 0, 0, 0, 1, 0, 0, 0]
  6: (10) [0, 0, 1, 1, 0, 1, 1, 0, 0, 0]
  7: (10) [0, 0, 0, 1, 1, 0, 1, 0, 1, 0]
  8: (10) [0, 0, 0, 0, 1, 0, 1, 0, 0, 0]
  9: (10) [0, 0, 1, 0, 1, 0, 0, 0, 1, 1]
  length: 10
```

## Пункт 4:



Степені вершин модифікованого графа:

Степінь вершини 0: 0(вхід) 2(вихід)

Степінь вершини 1: 1(вхід) 1(вихід)

Степінь вершини 2: 5(вхід) 3(вихід)

Степінь вершини 3: 4(вхід) 3(вихід)

Степінь вершини 4: 4(вхід) 2(вихід)

Степінь вершини 5: 1(вхід) 2(вихід)

Степінь вершини 6: 7(вхід) 4(вихід)

Степінь вершини 7: 1(вхід) 4(вихід)

Степінь вершини 8: 3(вхід) 2(вихід)

Степінь вершини 9: 1(вхід) 4(вихід)

## Усі шляхи довжини 2 і 3:

main.js:204 [0,3,3]

main.js:204 [0,3,4]

main.js:204 [0,3,6]

main.js:204 [0,8,4]

main.js:204 [0,8,6]

main.js:204 [1,6,2]

main.js:204 [1,6,3]

main.js:204 [1,6,5]

main.js:204 [1,6,6]

main.js:204 [2,1,6]

main.js:204 [2,2,1]

main.js:204 [2,2,2]

main.js:204 [2,2,6]

main.js:204 [2,6,2]

main.js:204 [2,6,3]

main.js:204 [2,6,5]

main.js:204 [2,6,6]

main.js:204 [3,3,3]

main.js:204 [3,3,4]

main.js:204 [3,3,6]

main.js:204 [3,4,2]

main.js:204 [3,4,7]

main.js:204 [3,6,2]

main.js:204 [3,6,3]

main.js:204 [3,6,5]

main.js:204 [3,6,6]  
main.js:204 [4,2,1]  
main.js:204 [4,2,2]  
main.js:204 [4,2,6]  
main.js:204 [4,7,3]  
main.js:204 [4,7,4]  
main.js:204 [4,7,6]  
main.js:204 [4,7,8]  
main.js:204 [5,2,1]  
main.js:204 [5,2,2]  
main.js:204 [5,2,6]  
main.js:204 [5,6,2]  
main.js:204 [5,6,3]  
main.js:204 [5,6,5]  
main.js:204 [5,6,6]  
main.js:204 [6,2,1]  
main.js:204 [6,2,2]  
main.js:204 [6,2,6]  
main.js:204 [6,3,3]  
main.js:204 [6,3,4]  
main.js:204 [6,3,6]  
main.js:204 [6,5,2]  
main.js:204 [6,5,6]  
main.js:204 [6,6,2]  
main.js:204 [6,6,3]  
main.js:204 [6,6,5]  
main.js:204 [6,6,6]  
main.js:204 [7,3,3]  
main.js:204 [7,3,4]  
main.js:204 [7,3,6]  
main.js:204 [7,4,2]  
main.js:204 [7,4,7]  
main.js:204 [7,6,2]  
main.js:204 [7,6,3]  
main.js:204 [7,6,5]  
main.js:204 [7,6,6]  
main.js:204 [7,8,4]

main.js:204 [7,8,6]  
main.js:204 [8,4,2]  
main.js:204 [8,4,7]  
main.js:204 [8,6,2]  
main.js:204 [8,6,3]  
main.js:204 [8,6,5]  
main.js:204 [8,6,6]  
main.js:204 [9,2,1]  
main.js:204 [9,2,2]  
main.js:204 [9,2,6]  
main.js:204 [9,4,2]  
main.js:204 [9,4,7]  
main.js:204 [9,8,4]  
main.js:204 [9,8,6]  
main.js:204 [9,9,2]  
main.js:204 [9,9,4]  
main.js:204 [9,9,8]  
main.js:204 [9,9,9]  
main.js:209 [0,3,3,3]  
main.js:209 [0,3,3,4]  
main.js:209 [0,3,3,6]  
main.js:209 [0,3,4,2]  
main.js:209 [0,3,4,7]  
main.js:209 [0,3,6,2]  
main.js:209 [0,3,6,3]  
main.js:209 [0,3,6,5]  
main.js:209 [0,3,6,6]  
main.js:209 [0,8,4,2]  
main.js:209 [0,8,4,7]  
main.js:209 [0,8,6,2]  
main.js:209 [0,8,6,3]  
main.js:209 [0,8,6,5]  
main.js:209 [0,8,6,6]  
main.js:209 [1,6,2,1]  
main.js:209 [1,6,2,2]  
main.js:209 [1,6,2,6]  
main.js:209 [1,6,3,3]

main.js:209 [1,6,3,4]  
main.js:209 [1,6,3,6]  
main.js:209 [1,6,5,2]  
main.js:209 [1,6,5,6]  
main.js:209 [1,6,6,2]  
main.js:209 [1,6,6,3]  
main.js:209 [1,6,6,5]  
main.js:209 [1,6,6,6]  
main.js:209 [2,1,6,2]  
main.js:209 [2,1,6,3]  
main.js:209 [2,1,6,5]  
main.js:209 [2,1,6,6]  
main.js:209 [2,2,1,6]  
main.js:209 [2,2,2,1]  
main.js:209 [2,2,2,2]  
main.js:209 [2,2,2,6]  
main.js:209 [2,2,6,2]  
main.js:209 [2,2,6,3]  
main.js:209 [2,2,6,5]  
main.js:209 [2,2,6,6]  
main.js:209 [2,6,2,1]  
main.js:209 [2,6,2,2]  
main.js:209 [2,6,2,6]  
main.js:209 [2,6,3,3]  
main.js:209 [2,6,3,4]  
main.js:209 [2,6,3,6]  
main.js:209 [2,6,5,2]  
main.js:209 [2,6,5,6]  
main.js:209 [2,6,6,2]  
main.js:209 [2,6,6,3]  
main.js:209 [2,6,6,5]  
main.js:209 [2,6,6,6]  
main.js:209 [3,3,3,3]  
main.js:209 [3,3,3,4]  
main.js:209 [3,3,3,6]  
main.js:209 [3,3,4,2]  
main.js:209 [3,3,4,7]

main.js:209 [3,3,6,2]  
main.js:209 [3,3,6,3]  
main.js:209 [3,3,6,5]  
main.js:209 [3,3,6,6]  
main.js:209 [3,4,2,1]  
main.js:209 [3,4,2,2]  
main.js:209 [3,4,2,6]  
main.js:209 [3,4,7,3]  
main.js:209 [3,4,7,4]  
main.js:209 [3,4,7,6]  
main.js:209 [3,4,7,8]  
main.js:209 [3,6,2,1]  
main.js:209 [3,6,2,2]  
main.js:209 [3,6,2,6]  
main.js:209 [3,6,3,3]  
main.js:209 [3,6,3,4]  
main.js:209 [3,6,3,6]  
main.js:209 [3,6,5,2]  
main.js:209 [3,6,5,6]  
main.js:209 [3,6,6,2]  
main.js:209 [3,6,6,3]  
main.js:209 [3,6,6,5]  
main.js:209 [3,6,6,6]  
main.js:209 [4,2,1,6]  
main.js:209 [4,2,2,1]  
main.js:209 [4,2,2,2]  
main.js:209 [4,2,2,6]  
main.js:209 [4,2,6,2]  
main.js:209 [4,2,6,3]  
main.js:209 [4,2,6,5]  
main.js:209 [4,2,6,6]  
main.js:209 [4,7,3,3]  
main.js:209 [4,7,3,4]  
main.js:209 [4,7,3,6]  
main.js:209 [4,7,4,2]  
main.js:209 [4,7,4,7]  
main.js:209 [4,7,6,2]

main.js:209 [4,7,6,3]

main.js:209 [4,7,6,5]

main.js:209 [4,7,6,6]

main.js:209 [4,7,8,4]

main.js:209 [4,7,8,6]

main.js:209 [5,2,1,6]

main.js:209 [5,2,2,1]

main.js:209 [5,2,2,2]

main.js:209 [5,2,2,6]

main.js:209 [5,2,6,2]

main.js:209 [5,2,6,3]

main.js:209 [5,2,6,5]

main.js:209 [5,2,6,6]

main.js:209 [5,6,2,1]

main.js:209 [5,6,2,2]

main.js:209 [5,6,2,6]

main.js:209 [5,6,3,3]

main.js:209 [5,6,3,4]

main.js:209 [5,6,3,6]

main.js:209 [5,6,5,2]

main.js:209 [5,6,5,6]

main.js:209 [5,6,6,2]

main.js:209 [5,6,6,3]

main.js:209 [5,6,6,5]

main.js:209 [5,6,6,6]

main.js:209 [6,2,1,6]

main.js:209 [6,2,2,1]

main.js:209 [6,2,2,2]

main.js:209 [6,2,2,6]

main.js:209 [6,2,6,2]

main.js:209 [6,2,6,3]

main.js:209 [6,2,6,5]

main.js:209 [6,2,6,6]

main.js:209 [6,3,3,3]

main.js:209 [6,3,3,4]

main.js:209 [6,3,3,6]

main.js:209 [6,3,4,2]

main.js:209 [6,3,4,7]

main.js:209 [6,3,6,2]

main.js:209 [6,3,6,3]

main.js:209 [6,3,6,5]

main.js:209 [6,3,6,6]

main.js:209 [6,5,2,1]

main.js:209 [6,5,2,2]

main.js:209 [6,5,2,6]

main.js:209 [6,5,6,2]

main.js:209 [6,5,6,3]

main.js:209 [6,5,6,5]

main.js:209 [6,5,6,6]

main.js:209 [6,6,2,1]

main.js:209 [6,6,2,2]

main.js:209 [6,6,2,6]

main.js:209 [6,6,3,3]

main.js:209 [6,6,3,4]

main.js:209 [6,6,3,6]

main.js:209 [6,6,5,2]

main.js:209 [6,6,5,6]

main.js:209 [6,6,6,2]

main.js:209 [6,6,6,3]

main.js:209 [6,6,6,5]

main.js:209 [6,6,6,6]

main.js:209 [7,3,3,3]

main.js:209 [7,3,3,4]

main.js:209 [7,3,3,6]

main.js:209 [7,3,4,2]

main.js:209 [7,3,4,7]

main.js:209 [7,3,6,2]

main.js:209 [7,3,6,3]

main.js:209 [7,3,6,5]

main.js:209 [7,3,6,6]

main.js:209 [7,4,2,1]

main.js:209 [7,4,2,2]

main.js:209 [7,4,2,6]

main.js:209 [7,4,7,3]

main.js:209 [7,4,7,4]  
main.js:209 [7,4,7,6]  
main.js:209 [7,4,7,8]  
main.js:209 [7,6,2,1]  
main.js:209 [7,6,2,2]  
main.js:209 [7,6,2,6]  
main.js:209 [7,6,3,3]  
main.js:209 [7,6,3,4]  
main.js:209 [7,6,3,6]  
main.js:209 [7,6,5,2]  
main.js:209 [7,6,5,6]  
main.js:209 [7,6,6,2]  
main.js:209 [7,6,6,3]  
main.js:209 [7,6,6,5]  
main.js:209 [7,6,6,6]  
main.js:209 [7,8,4,2]  
main.js:209 [7,8,4,7]  
main.js:209 [7,8,6,2]  
main.js:209 [7,8,6,3]  
main.js:209 [7,8,6,5]  
main.js:209 [7,8,6,6]  
main.js:209 [8,4,2,1]  
main.js:209 [8,4,2,2]  
main.js:209 [8,4,2,6]  
main.js:209 [8,4,7,3]  
main.js:209 [8,4,7,4]  
main.js:209 [8,4,7,6]  
main.js:209 [8,4,7,8]  
main.js:209 [8,6,2,1]  
main.js:209 [8,6,2,2]  
main.js:209 [8,6,2,6]  
main.js:209 [8,6,3,3]  
main.js:209 [8,6,3,4]  
main.js:209 [8,6,3,6]  
main.js:209 [8,6,5,2]  
main.js:209 [8,6,5,6]  
main.js:209 [8,6,6,2]



main.js:209 [8,6,6,3]  
main.js:209 [8,6,6,5]  
main.js:209 [8,6,6,6]  
main.js:209 [9,2,1,6]  
main.js:209 [9,2,2,1]  
main.js:209 [9,2,2,2]  
main.js:209 [9,2,2,6]  
main.js:209 [9,2,6,2]  
main.js:209 [9,2,6,3]  
main.js:209 [9,2,6,5]  
main.js:209 [9,2,6,6]  
main.js:209 [9,4,2,1]  
main.js:209 [9,4,2,2]  
main.js:209 [9,4,2,6]  
main.js:209 [9,4,7,3]  
main.js:209 [9,4,7,4]  
main.js:209 [9,4,7,6]  
main.js:209 [9,4,7,8]  
main.js:209 [9,8,4,2]  
main.js:209 [9,8,4,7]  
main.js:209 [9,8,6,2]  
main.js:209 [9,8,6,3]  
main.js:209 [9,8,6,5]  
main.js:209 [9,8,6,6]  
main.js:209 [9,9,2,1]  
main.js:209 [9,9,2,2]  
main.js:209 [9,9,2,6]  
main.js:209 [9,9,4,2]  
main.js:209 [9,9,4,7]  
main.js:209 [9,9,8,4]  
main.js:209 [9,9,8,6]  
main.js:209 [9,9,9,2]  
main.js:209 [9,9,9,4]  
main.js:209 [9,9,9,8]

Матриця досяжності:

```

(10) [Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10)]
  0: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  1: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  2: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  3: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  4: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  5: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  6: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  7: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  8: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  9: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
    length: 10
  [[Prototype]]: Array(0)

```

Матриця сильної зв'язності:

```

(10) [Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10), Array(10)]
  0: (10) [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  1: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  2: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  3: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  4: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  5: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  6: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  7: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  8: (10) [0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
  9: (10) [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    length: 10
  [[Prototype]]: Array(0)

```

Перелік КСЗ:

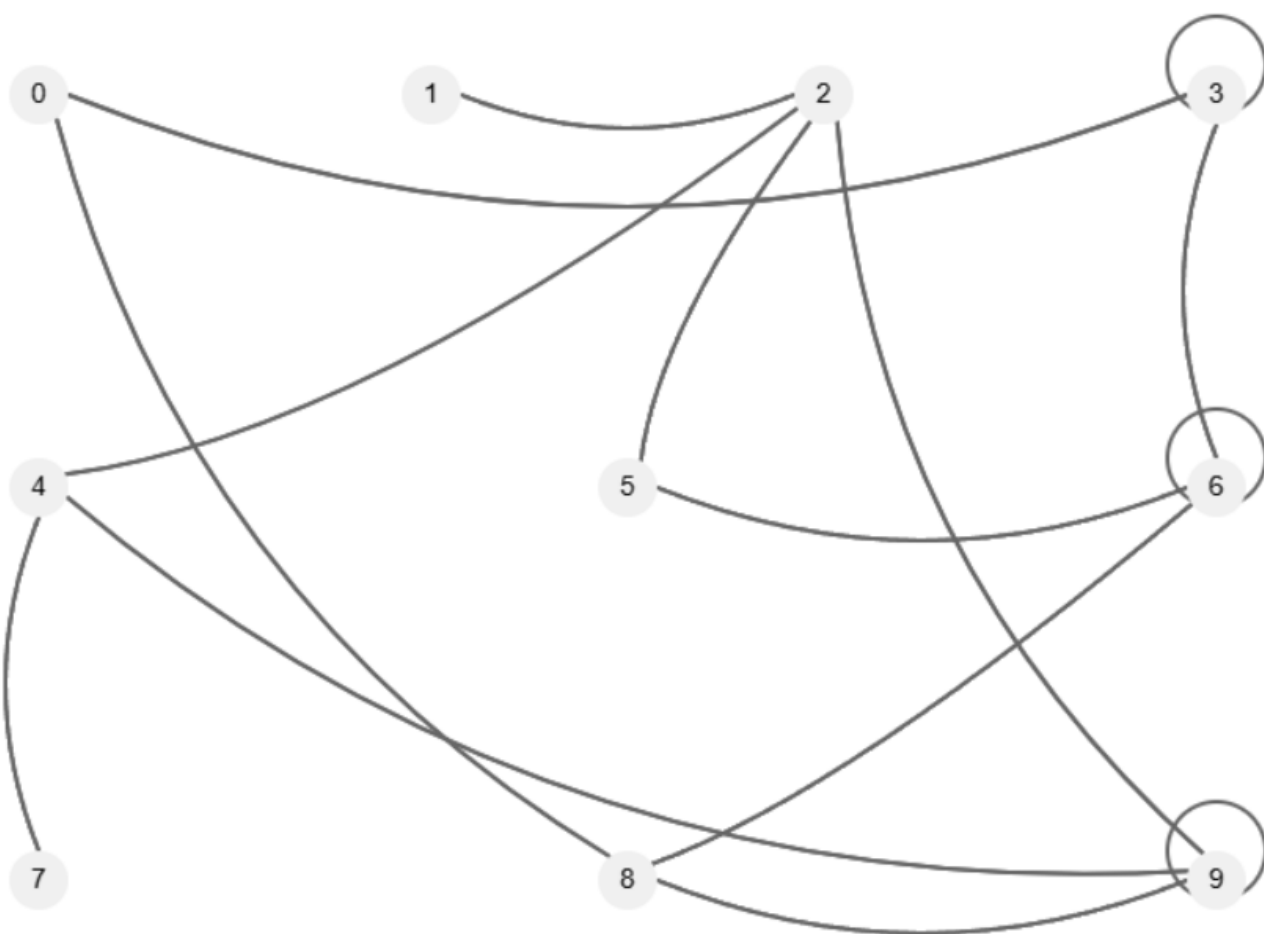
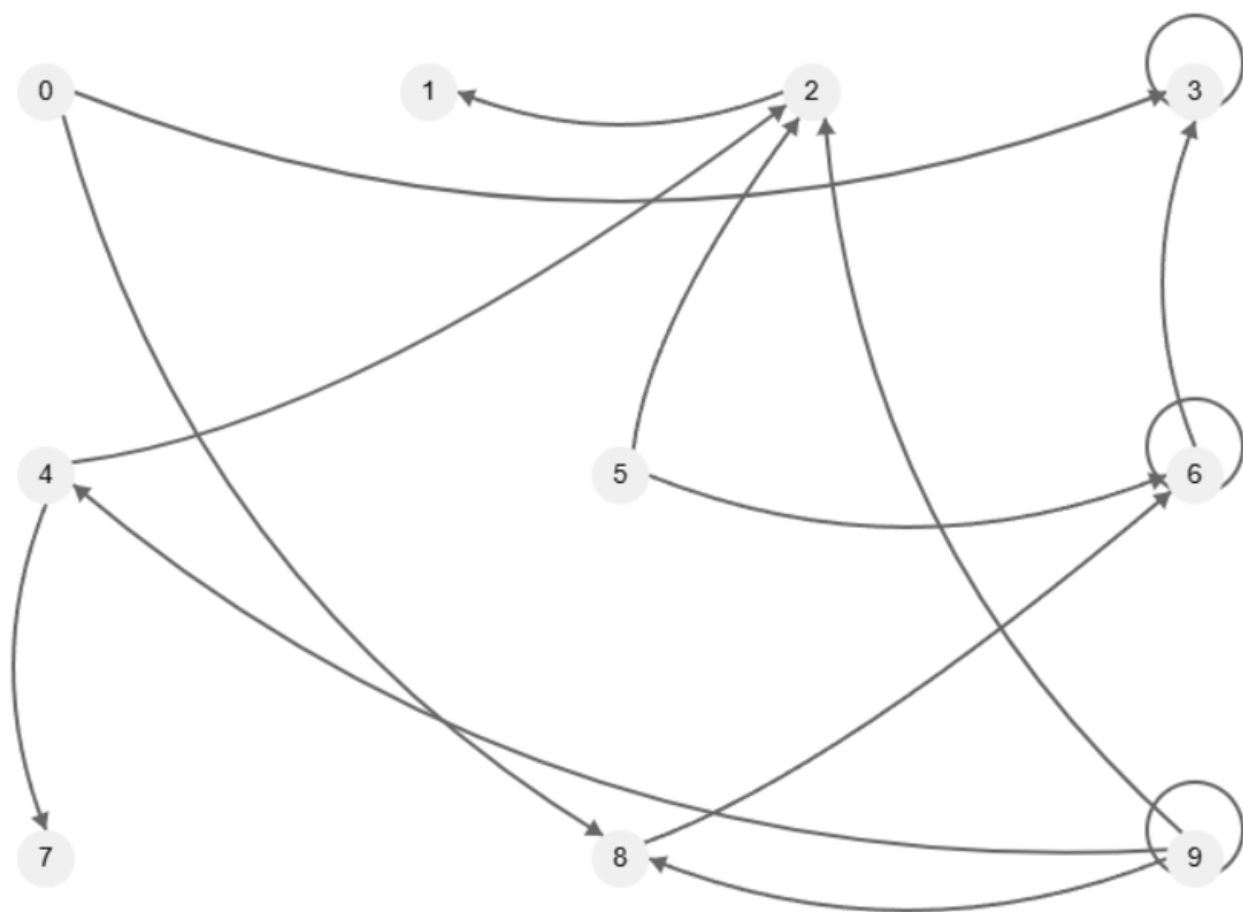
```

(3) [Array(1), Array(8), Array(1)]
  0: [0]
  1: (8) [1, 2, 3, 4, 5, 6, 7, 8]
  2: [9]

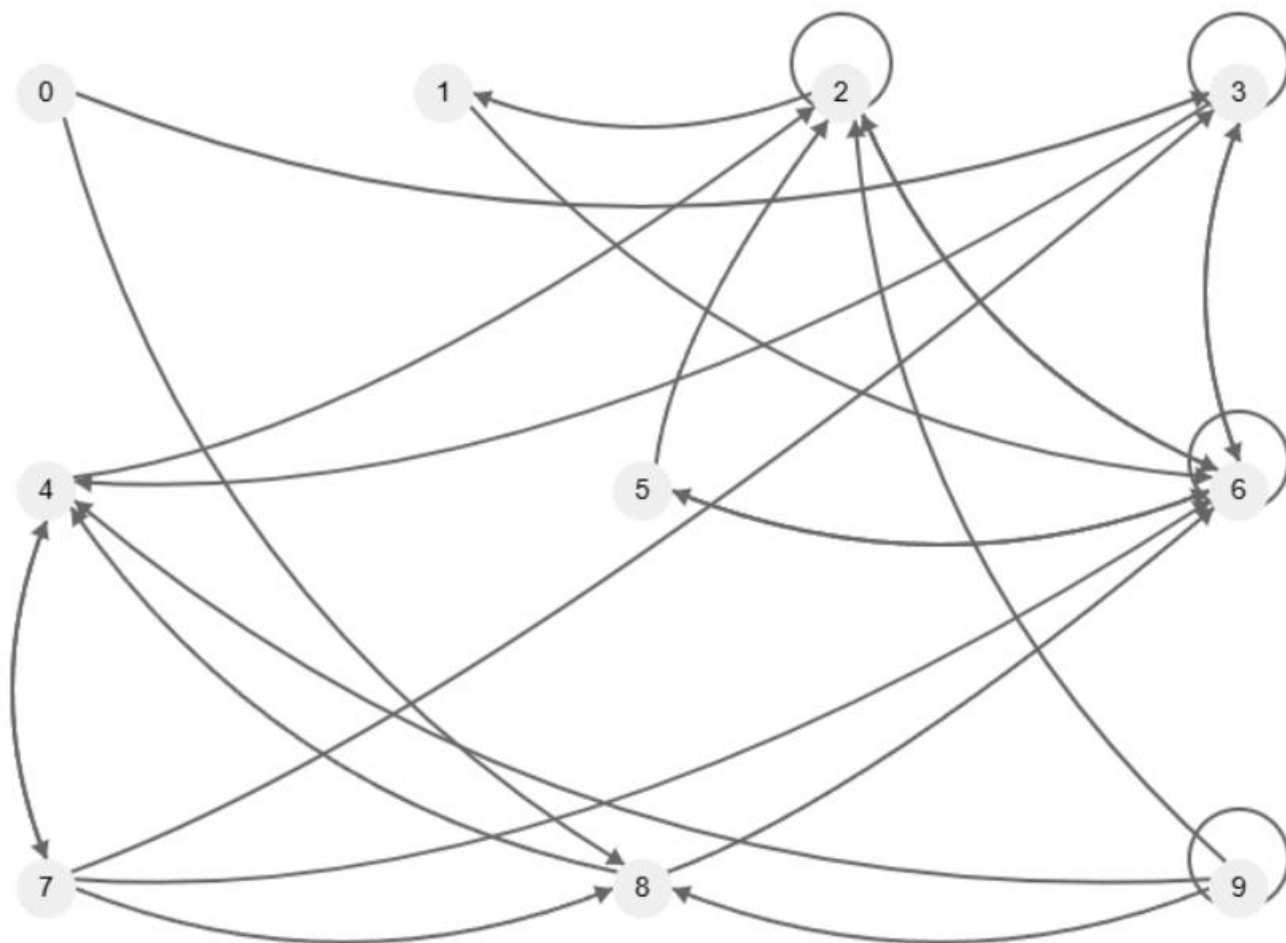
```

**Скриншоти:**

**Напрямлений і ненапрямлений графи:**



**Модифікований граф з іншим коефіцієнтом:**



### Граф конденсації:



### Висновок:

Під час цієї лабораторної я вивчив багато чого. Перше, я повторив і вивчив багато нового з теорії графів, усі визначення що були використані у цій лабораторній. Друге, навчився робити з вкладеними списками і множинами у JS, як робити по ним обходи і створювати такі на льоту. Цікаво було дивитися на граф зі сторони матриці і робити з ним різні операції. На додачу, повторив різні базові речі - як написання функцій, переведення списків у стрічки і порівнювання їх таким чином (бо стандартно у JS порівнюються посилання), ітерації, елемент `<canvas>` у HTML і багато чого ще