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 №3

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

Постановка задачі

1. Представити у програмі напрямлений і ненапрямлений графи з заданими параметрами:
   - кількість вершин n;
   - розміщення вершин;
   - матриця суміжності A.
2. Створити програму для формування зображення напрямленого і ненапрямленого графів у графічному вікні.

Мій сід був 4108. На жаль, у JavaScript немає вбудованого генератора сідом, тому я згенерував константу таблицю для перевірки і одну таблицю генерую у коді повністю випадково. Ось моя константа таблиця:

```
const predeterminedArray = [
[0, 1, 0, 1, 0, 1, 1, 0, 0, 1],
[1, 1, 1, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 1, 0, 1, 1, 0, 1, 0, 0],
[1, 0, 0, 1, 0, 1, 1, 1, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 1, 0, 1],
[0, 0, 1, 0, 0, 1, 0, 0, 0, 0],
[0, 1, 0, 0, 1, 0, 0, 0, 0, 1],
[0, 0, 1, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 1, 0, 0],
];
```

Текст програми:

HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Лабораторна 3</title>
  <style>
    canvas { border: 1px solid #000000; }
  </style>
</head>
<body>
  <canvas id="canvasDetermined" width="800" height="600"></canvas>
  <canvas id="canvasRandom" width="800" height="600"></canvas>
  <script src="main.js"></script>
</body>
</html>
```

JS:

```
const canvasD = document.getElementById('canvasDetermined');
```

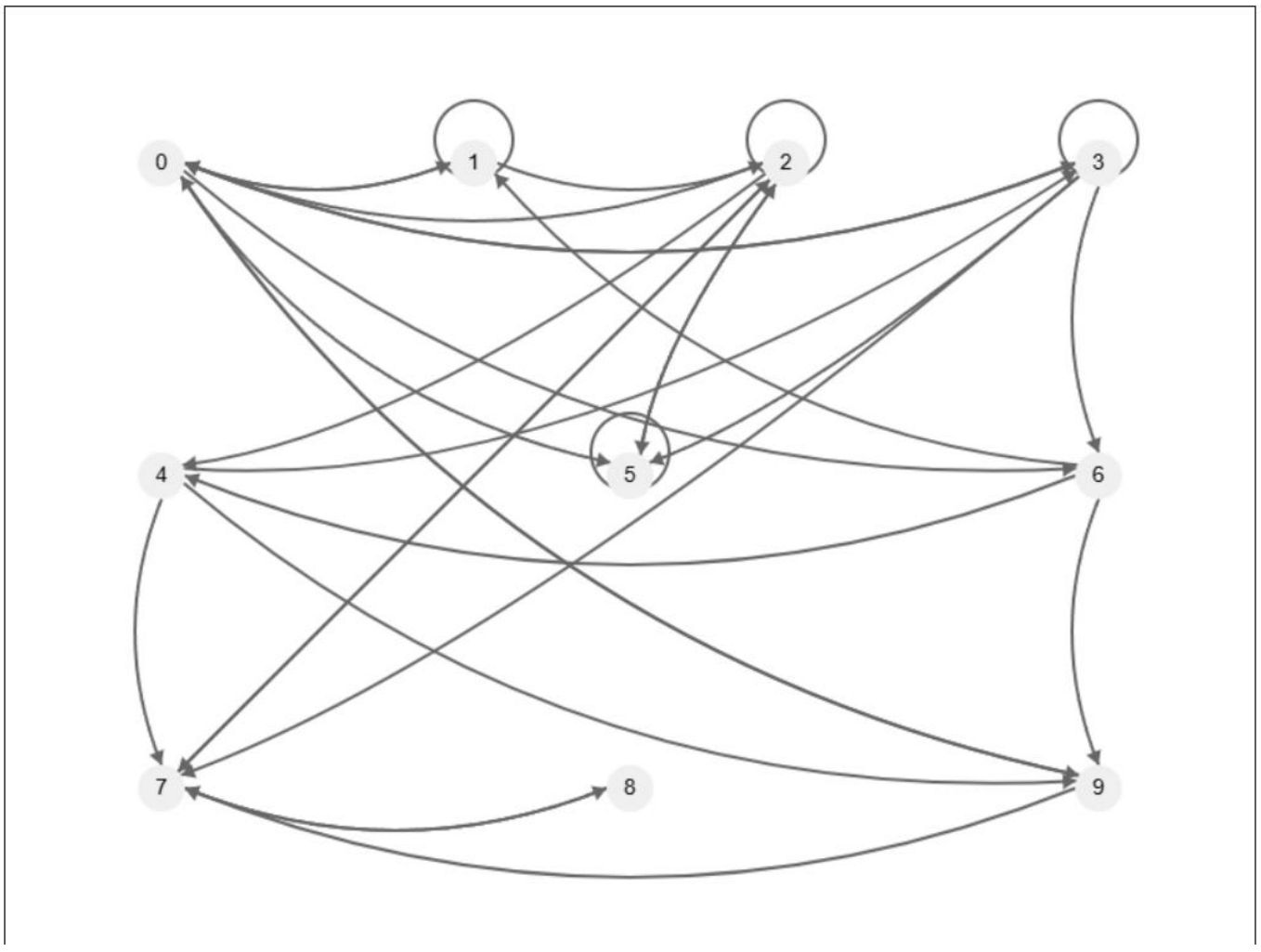
```
const canvasR = document.getElementById('canvasRandom');
const n1 = 4;
const n2 = 1;
const n3 = 0;
const n4 = 8;
const k = 1 - (n3 * 0.02) - (n4 * 0.005) - 0.25;
const nodeNumber = 10 + n3;
const nodeRadius = 15;
const nodes = [
    {x: 100, y: 100},
    {x: 300, y: 100},
    {x: 500, y: 100},
    {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 predeterminedArray = [
[0, 1, 0, 1, 0, 1, 1, 0, 0, 1],
[1, 1, 1, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 1, 0, 1, 1, 0, 1, 0, 0],
[1, 0, 0, 1, 0, 1, 1, 1, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 1, 0, 1],
[0, 0, 1, 0, 0, 1, 0, 0, 0, 0],
[0, 1, 0, 0, 1, 0, 0, 0, 0, 1],
[0, 0, 1, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 1, 0, 0],
];
// на жаль, у Джаваскріпт не вбудовано сідованого генератора, тому я додав визначену матрицю для демонстрації
const randomizeArray = () => {
const result = [];
for (let i = 0; i < nodeNumber; i++) {
    const row = [];
    for (let j = 0; j < nodeNumber; j++) {
    row.push(Math.floor(Math.random() * 2 * k));
    }
    result.push(row);
} return result };
const randomArray = randomizeArray();
console.log('Predetermined Array:');
console.log(predeterminedArray);
console.log('Random Array:');
console.log(randomArray);

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';
```

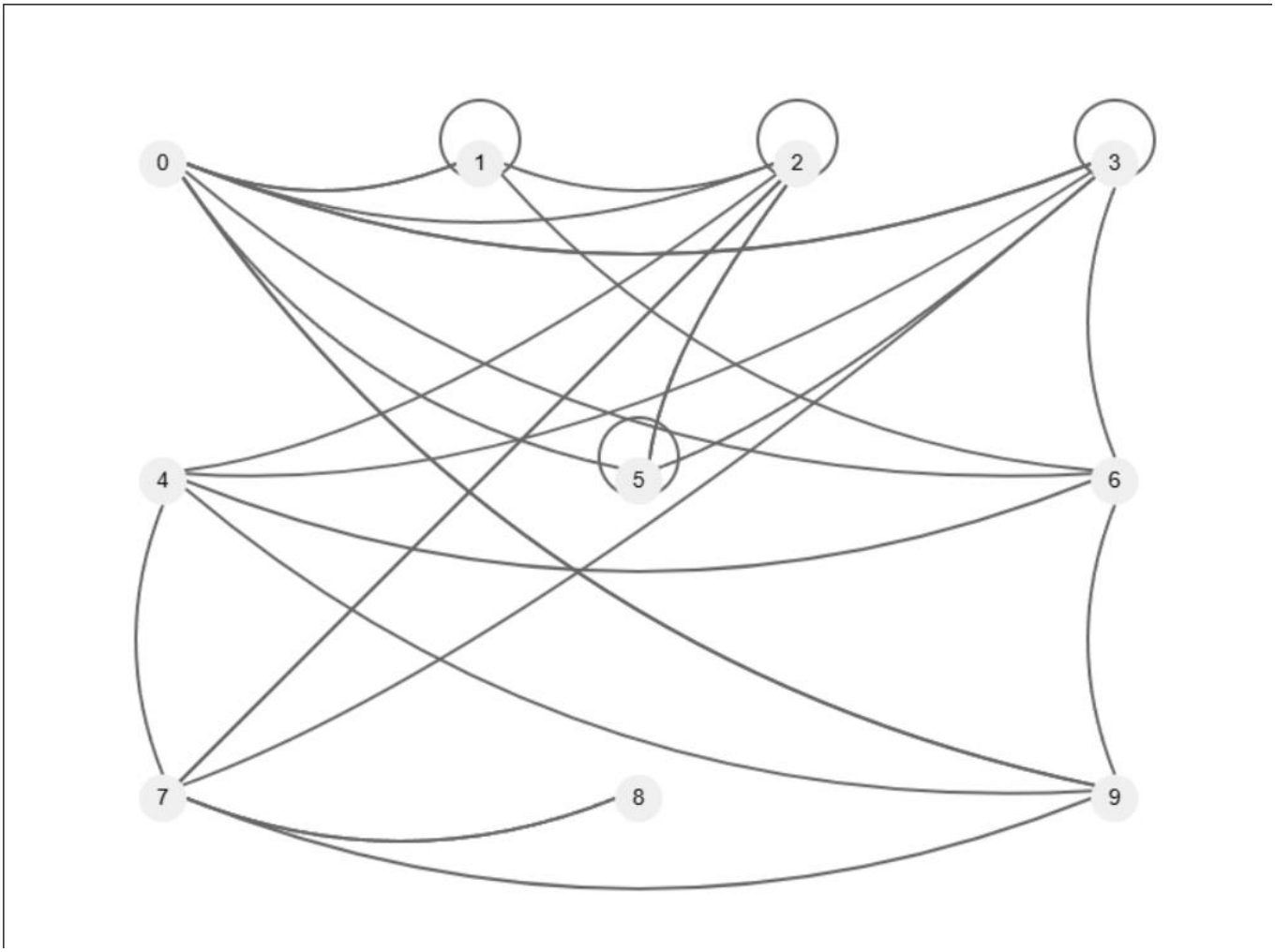
```
    ctx.fill();
    ctx.fillStyle = '#000';
    ctx.font = '14px Arial';
    ctx.textAlign = 'center';
    ctx.textBaseline = 'middle';
    ctx.fillText(id, node.x, node.y);
  });
}

drawGraph(canvasD, nodes, predeterminedArray, true);
drawGraph(canvasR, nodes, randomArray, true);
```

Тестування програми:

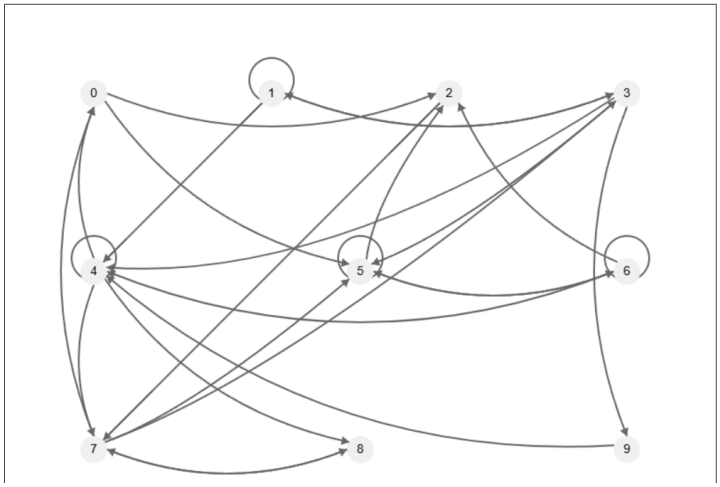
Напрямлений граф по константній матриці:

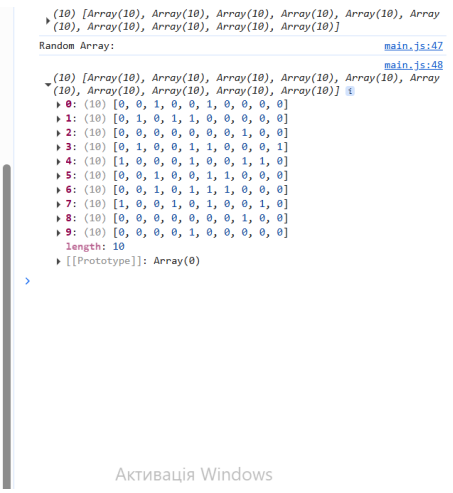


Ненапрямлений граф по константній матриці:



Випадковий згенерований напрямлений граф:





Висновок:

У результаті роботи я вивчив роботу з елементом <canvas> у HTML, як малювати у ньому криві, і зобразив завдяки ньому напрямлені і ненапрямлені графи. Також я навчився з матриці з’єднаннь отримувати сам граф і повторив генерацію випадкових чисел за допомогою Math.random(). Впевнений, цей досвід знадобиться мені у наступних лабораторних роботах