

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики

Мегафакультет компьютерных технологий и управления

Кафедра информатики и прикладной математики



ITMO UNIVERSITY

Алгоритмы и структуры данных

Лабораторная работа №4

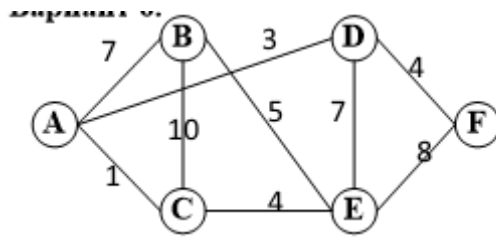
«Потоки в сетях»

Группа: P3218

Студент: Петкевич Константин

Преподаватель: Зинчик А. А.

2017г



Граф с расставленными пропускными способностями.

Минимальный разрез данного графа - { AB, AD, AC } - сумма его пропускных способностей = 11.

Листинг

```

class FordFulkerson {
    private int size;
    private int[][] graph;
    public FordFulkerson(int V, int[][] graph) {
        this.size = V;
        this.graph = graph;
    }

    private boolean unfilledPathExists(int rGraph[][], int source, int destination, int parent[]) {

        boolean visited[] = new boolean[size];
        for(int i = 0; i < size; ++i) {
            visited[i] = false;
        }

        LinkedList<Integer> queue = new LinkedList<Integer>();
        queue.add(source);
        visited[source] = true;
        parent[source] = -1;

        while (queue.size() != 0) {
            int u = queue.poll();

            for (int vertexId = 0; vertexId < size; vertexId++) {
                if (visited[vertexId] == false && rGraph[u][vertexId] > 0) {
                    queue.add(vertexId);
                    parent[vertexId] = u;
                    visited[vertexId] = true;
                }
            }
        }

        return (visited[destination] == true);
    }

    int solve(int source, int destination) {
        int u, curVertex;

        int rGraph[][] = new int[size][size];

        for (u = 0; u < size; u++)
            for (curVertex = 0; curVertex < size; curVertex++)
                rGraph[u][curVertex] = graph[u][curVertex];

        int parent[] = new int[size];

        int max_flow = 0;

        while (unfilledPathExists(rGraph, source, destination, parent)) {
            int path_flow = Integer.MAX_VALUE;

            for (curVertex = destination; curVertex != source; curVertex = parent[curVertex]) {

```

```

    u = parent[curVertex];
    path_flow = Math.min(path_flow, rGraph[u][curVertex]);
}

for (curVertex = destination; curVertex != source; curVertex = parent[curVertex]) {
    u = parent[curVertex];
    rGraph[u][curVertex] -= path_flow;
    rGraph[curVertex][u] += path_flow;
}

max_flow += path_flow;
}

return max_flow;
}

public static void main (String[] args) throws java.lang.Exception {
    int graph[][] = new int[][] {
        {0, 7, 3, 1, 0, 0},
        {0, 0, 0, 7, 0, 4},
        {0, 0, 0, 10, 5, 0},
        {0, 0, 0, 0, 4, 0},
        {0, 0, 0, 0, 0, 8},
        {0, 0, 0, 0, 0, 0}
    };

    FordFulkerson m = new FordFulkerson(6, graph);

    System.out.println(m.solve(0, 5));
}
}

```

Вывод: теорема Форда-Фалкерсона справедлива - максимальный поток, который нашел алгоритм равен найденному мною минимальному разрезу данной сети.