|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ДОМАШНЕГО ЗАДАНИЯ**

Студент Мягкова Анастасия Сергеевна

Группа РК6-81Б

Тип задания Домашнее задание

Тема домашнего задания Устройства ЭВМ

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Мягкова А.С.\_**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Берчун Ю.В.\_**

*подпись, дата фамилия, и.о.*

*Москва, 2019 г.*

Оглавление

[Задание на лабораторную работу 3](#_Toc6846160)

[Последовательный алгоритм размещения 4](#_Toc6846161)

[Итерационный этап решения 4](#_Toc6846162)

[Реализация задачи 5](#_Toc6846163)

# Задание на лабораторную работу

Реализовать программу, решающую задачу компоновки при конструировании ЭВМ в соответствии с матрицей смежности, отображающей количество связей между элементами (строки и столбцы матрицы — элементы будущей схемы; значения матрицы — количество связей)

Матрица смежности:

0, 0, 3, 0, 2, 0, 0, 0, 2, 3, 0, 1, 4, 0, 3, 2, 2, 0, 4, 0, 4, 2, 2, 0, 0, 0, 0, 0, 2, 2

0, 0, 1, 0, 2, 0, 0, 4, 0, 2, 0, 4, 1, 0, 3, 1, 0, 0, 4, 4, 0, 1, 0, 3, 3, 0, 0, 0, 0, 4

3, 1, 0, 0, 2, 2, 3, 3, 0, 0, 4, 0, 3, 0, 4, 2, 3, 0, 0, 0, 0, 0, 4, 1, 4, 4, 3, 0, 0, 0

0, 0, 0, 0, 0, 1, 0, 4, 4, 0, 4, 1, 0, 0, 1, 0, 0, 3, 4, 2, 3, 0, 0, 3, 0, 0, 0, 0, 1, 2

2, 2, 2, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 4, 0, 0, 0, 0, 4, 3, 3, 4, 0, 4, 0, 4, 4

0, 0, 2, 1, 1, 0, 4, 0, 1, 3, 0, 0, 0, 3, 0, 0, 2, 0, 0, 1, 0, 0, 1, 1, 3, 0, 0, 3, 0, 0

0, 0, 3, 0, 2, 4, 0, 3, 4, 1, 0, 0, 0, 3, 0, 4, 0, 0, 0, 4, 0, 0, 3, 0, 1, 0, 4, 0, 3, 0

0, 4, 3, 4, 0, 0, 3, 0, 0, 3, 0, 4, 3, 1, 0, 0, 0, 0, 1, 3, 0, 4, 0, 3, 3, 2, 1, 0, 3, 0

2, 0, 0, 4, 0, 1, 4, 0, 0, 2, 1, 0, 2, 3, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 4, 0, 0, 4, 1, 1

3, 2, 0, 0, 0, 3, 1, 3, 2, 0, 2, 0, 1, 0, 0, 3, 0, 0, 1, 0, 0, 3, 0, 2, 0, 0, 0, 3, 0, 1

0, 0, 4, 4, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 3, 0, 4, 0, 0, 0, 4, 0, 0, 1, 0, 0

1, 4, 0, 1, 0, 0, 0, 4, 0, 0, 0, 0, 0, 4, 3, 0, 2, 2, 2, 0, 0, 1, 4, 0, 0, 2, 3, 0, 0, 0

4, 1, 3, 0, 0, 0, 0, 3, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 0, 1, 0, 0, 0, 0, 0, 0

0, 0, 0, 0, 0, 3, 3, 1, 3, 0, 0, 4, 0, 0, 2, 0, 0, 0, 3, 0, 0, 3, 0, 3, 3, 0, 0, 4, 3, 1

3, 3, 4, 1, 1, 0, 0, 0, 1, 0, 0, 3, 0, 2, 0, 1, 2, 4, 0, 3, 2, 2, 4, 0, 2, 0, 0, 0, 0, 0

2, 1, 2, 0, 0, 0, 4, 0, 0, 3, 0, 0, 1, 0, 1, 0, 0, 2, 0, 1, 1, 1, 0, 4, 0, 2, 1, 1, 1, 3

2, 0, 3, 0, 4, 2, 0, 0, 0, 0, 1, 2, 0, 0, 2, 0, 0, 2, 1, 0, 1, 0, 3, 3, 4, 2, 3, 0, 1, 0

0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 4, 2, 2, 0, 0, 0, 2, 0, 3, 1, 4, 1, 0, 0, 0, 1

4, 4, 0, 4, 0, 0, 0, 1, 0, 1, 3, 2, 0, 3, 0, 0, 1, 0, 0, 2, 4, 3, 0, 0, 4, 1, 2, 2, 2, 0

0, 4, 0, 2, 0, 1, 4, 3, 2, 0, 0, 0, 0, 0, 3, 1, 0, 0, 2, 0, 3, 0, 1, 1, 0, 0, 0, 4, 2, 0

4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 2, 1, 1, 2, 4, 3, 0, 0, 4, 1, 0, 1, 1, 4, 0, 0

2, 1, 0, 0, 4, 0, 0, 4, 0, 3, 0, 1, 3, 3, 2, 1, 0, 0, 3, 0, 0, 0, 0, 1, 1, 0, 2, 0, 0, 0

2, 0, 4, 0, 3, 1, 3, 0, 0, 0, 0, 4, 0, 0, 4, 0, 3, 3, 0, 1, 4, 0, 0, 1, 4, 0, 0, 0, 4, 0

0, 3, 1, 3, 3, 1, 0, 3, 0, 2, 0, 0, 1, 3, 0, 4, 3, 1, 0, 1, 1, 1, 1, 0, 0, 2, 0, 2, 0, 0

0, 3, 4, 0, 4, 3, 1, 3, 4, 0, 4, 0, 0, 3, 2, 0, 4, 4, 4, 0, 0, 1, 4, 0, 0, 1, 3, 2, 0, 3

0, 0, 4, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 2, 1, 1, 0, 1, 0, 0, 2, 1, 0, 4, 0, 1, 0

0, 0, 3, 0, 4, 0, 4, 1, 0, 0, 0, 3, 0, 0, 0, 1, 3, 0, 2, 0, 1, 2, 0, 0, 3, 4, 0, 3, 0, 3

0, 0, 0, 0, 0, 3, 0, 0, 4, 3, 1, 0, 0, 4, 0, 1, 0, 0, 2, 4, 4, 0, 0, 2, 2, 0, 3, 0, 2, 0

2, 0, 0, 1, 4, 0, 3, 3, 1, 0, 0, 0, 0, 3, 0, 1, 1, 0, 2, 2, 0, 0, 4, 0, 0, 1, 0, 2, 0, 0

2, 4, 0, 2, 4, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 3, 0, 1, 0, 0, 0, 0, 0, 0, 3, 0, 3, 0, 0, 0

# Последовательный алгоритм размещения

1. Сначала необходимо найти вершину с минимальной суммой по строке S.

Сумма элементов в *i*-той строке, где *n* – размерность матрицы

1. Далее находим инцидентные вершины, к найденной ранее вершине с минимальной суммой по строке, и добавляем их в текущую группу.
2. В случае если количество вершин меньше, чем необходимо, нужно добавить вершины, которые инцидентны вершинам группы.
3. Если количество вершин больше, то считается дельта функция, и элемент с наибольшим значением функции исключается из группы.

Где – группа кандидатов

1. Добавляем созданную группу

# Итерационный этап решения

Для второго этапа необходимо опорное решение, полученное в результате выполнения алгоритма размещения.

1. Чтобы получить оптимальное размещение вершин по группам, необходимо посчитать матрицу перестановочных коэффициентов для каждой подгруппы:
2. В матрице находится максимальный элемент. Его индексы соответствуют порядковым номерам вершин, которые необходимо поменять местами.
3. В случае, если , перегруппировка для данной группы заканчивается

# Реализация задачи

Программа была реализована на языке JavaScript. Были выполнены следующие разбиения:

* 6 групп по 5 элементов
* 5 групп по 6 элементов
* 4 группы по 7 элементов и 1 группа по 2

Итоговые варианты компоновки для каждого из проверенных вариантов представлены ниже.

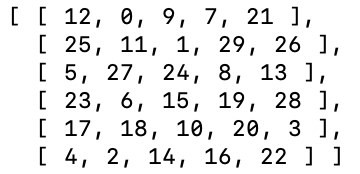


Рисунок 1. Разбиение 6 по 5

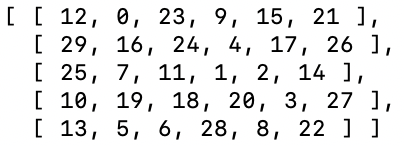


Рисунок 2. Разбиение 5 по 6

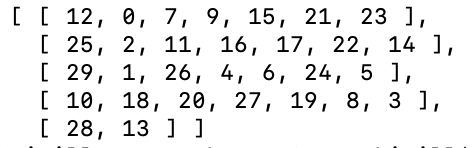


Рисунок 3. Разбиение 4 по 7, 1 по 2