**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский

технический университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

Кафедра систем автоматизированного проектирования

09.03.01. «Информатика и вычислительная техника»

Курсовая работа

по дисциплине: «Методы программирования систем автоматизированного проектирования»  
на тему: «Решение задачи о размещении вершин графа в линейке с использованием метода ветвей и границ»

Обучающийся 4215 \_\_\_\_\_\_\_\_\_\_\_\_ Закиров И.Р.

(номер группы) (подпись, дата) (Ф.И.О.)

Руководитель старший преподаватель Суздальцев И.В.

(должность) (Ф.И.О.)

Курсовая работа (проект) зачтена (зачтен) с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Казань 2020

**СОДЕРЖАТЕЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ**

В этой работе будет рассматриваться задача размещения графа на линейке. В самой задаче решение ищется для взвешенного графа. Граф G (V, U) – математический объект, являющийся совокупностью множеств вершин V и рёбер U. Каждое ребро может соединять только две вершины. Взвешенный граф — граф, каждому ребру которого поставлено в соответствие некое значение (вес ребра). Обычно вес – это число, и его интерпретируют как длину ребра.

Исходные данные для решения задачи: количество вершин графа, матрица смежности с указанием веса рёбер между вершинами. Матрица смежности – матрица элемент которого равен весу ребра, соединяющего вершины и . Далее, на рис. 1, представлен пример графа и его размещения на линейке:

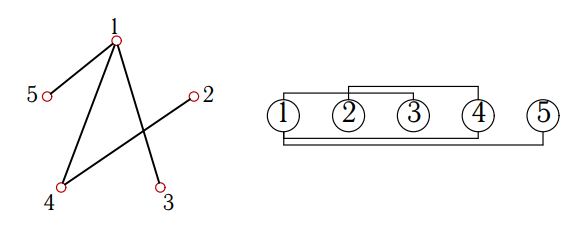


Рис. 1. Невзвешенный граф и его разложение на линейке

В результате мы должны получить оптимальные позиции вершин графа на линейке с минимальной суммарной длиной рёбер. Выводиться будет в виде n-ки вершин, расположенных в линейке (например, [3, 5, 2, 4, 1]).

Задача размещения графа на линейке (а так же на плоскости) имеет реальное практическое применение: размещение микросхем с использованием наименьшего количества соединительного материала. Это полезно и экономией материала, и улучшением пропускной способности соединений. Так же проблема размещения, исходя из названия, может возникнуть при размещении чего угодно, у чего есть межэлементные связи и рекомендации эти связи оптимизировать. Прецеденты можно найти на множестве предприятий, к примеру, при работе с конвейерами.

**МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ**

Таблица 1. Обозначения элементов математической модели задачи

|  |  |  |
| --- | --- | --- |
| **Обозначения элементов модели** | **Описание элементов модели** | **Примечание** |
|  | Исходный граф |  |
|  | Множество вершин |  |
|  | Множество рёбер |  |
|  | Количество вершин графа |  |
|  | Количество позиций |  |
|  | Матрица смежности с указанием веса рёбер между вершинами |  |
|  | Вес ребра, инцидентного -й и -й вершинам графа | *,* , |
|  | Кортеж, определяющий одно из возможных решений |  |
|  | Вершина графа, размещённая в данной позиции |  |

Таблица 1 содержит все обозначения, используемые в математической постановке задачи.

Целевая функция:

Ограничения задачи:

В решении для каждой вершины назначается только одна позиция (m = n).

Количество потенциальных решений для задачи о размещении графа определяется по комбинаторной формуле (как количество перестановок вершин графа):

где – количество вершин графа, – общее число вариантов размещения вершин графа на линейке.

**ОПИСАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ**

**Общая информация**

Метод ветвей и границ – метод, предназначенный для нахождения решений задач оптимизации. Метод является развитием метода полного перебора, он позволяет отсеивать часть возможных неоптимальных решений. Впервые был предложен в 1960-м году Алисой Лэнд и Элисон Дойг. Распространён для решения NP-сложных задач на оптимизацию.

Основа метода – сочетание двух действий: последовательное разбиение множества решений на подмножества (ветвление) и оценка верхней или нижней границы и отбрасывание бесперспективных ветвей. На каждом шаге метода элементы разбиения подвергаются проверке для выяснения, содержит данное подмножество оптимальное решение или нет. Проверка осуществляется посредством вычисления оценки снизу для целевой функции на данном подмножестве. Если оценка снизу не меньше рекорда — наилучшего из найденных решений, то подмножество может быть отброшено. Проверяемое подмножество может быть отброшено еще и в том случае, когда в нем удается найти наилучшее решение. Если значение целевой функции на найденном решении меньше рекорда, то происходит смена рекорда. По окончанию работы алгоритма рекорд является результатом его работы. Если удается отбросить все элементы разбиения, то рекорд — оптимальное решение задачи. В противном случае, из неотброшенных подмножеств выбирается наиболее перспективное (например, с наименьшим значением нижней оценки), и оно подвергается разбиению. Новые подмножества вновь подвергаются проверке и т.д. Вычисление нижней границы является важнейшим элементом данной схемы.

**Описание шагов алгоритма**

R – матрица смежности графа.

D – матрица расстояний между позициями.

n – количество вершин в графе.

Fq – суммарная длина соединений между размещёнными элементами.

wP – длина соединений между размещёнными и неразмещёнными элементами.

vP – длина соединений между неразмещёнными элементам.

minF – минимальная нижняя граница целевой функции на данном этапе.

minP – позиция, с которой связана minF.

1. Инициализация: создаём список Р из значений от 1 до n. Создаём пустой список Pp для занятых позиций из Р. Первое значение будет указывать, на какой позиции находится первая вершина, второе значение – местонахождение второй вершины и так далее. Создаём переменную Ei, изначально равную единице, обозначающую, какая вершина в данный момент размещается. Создаём переменные FP, Fq, wP, vP, minF, minP.

2. В minF записываем максимально возможное значение типа данных.

3. Начинаем проходится по списку P, перебирая все значения (шаги 3-8).

4. Обнуляем FP, Fq, wP, vP.

5. Если Pp не пуст, то:

5.1. От 1 до Ei включительно находим (по матрице R) соединённые рёбрами вершины и вычисляем Fq, складывая перемноженные веса рёбер между вершинами и расстояния между их размещениями (местоположения вершин от 1 до Ei-1 находятся в списке Pp, вершине Ei соответствует текущее значение из P).

Иначе:

5.2. Fq = 0.

6. Находим wP. Для этого перебираем вершины от 1 до Ei включительно, для каждой находя по матрице R расстояния от них самих до вершин от Ei+1 до n включительно. Значения эти записываем в вектор r в порядке не возрастания. Так же для каждой занятой, из списка Pp, (и одной занимаемой в данный момент) позиции находим по матрице D расстояния до незанятых позиций (перебираем для этого P) и записываем эти значения в вектор d в порядке не убывания. Скалярно перемножаем вектора r и d для нахождения wP.

7. Находим длину vP. Находим по матрице R пересечения вершин от Ei+1 до n включительно друг с другом, образуя невозрастающий вектор r, а по матрице D находим пересечения позиций из P друг с другом, образуя неубывающий вектор d. Скалярно перемножаем вектора r и d для нахождения vP.

Вычисляем FP, складывая Fq, wP и vP.

8. Если FP меньше minF, то перезаписываем minF=FP и minP = текущий элемент P (8.1).

9. Перебираем список P, убирая из него значение minP и добавляя его в конец списка Pp (9.2).

10. Увеличиваем Ei на 1.

11. Повторяем шаги 2-10, пока список Р не пуст.

12. Выводим искомое размещение, создав массив, в котором каждому элементу будет соответствовать индекс соответствующей позиции в P.

**Алгоритмическая схема**

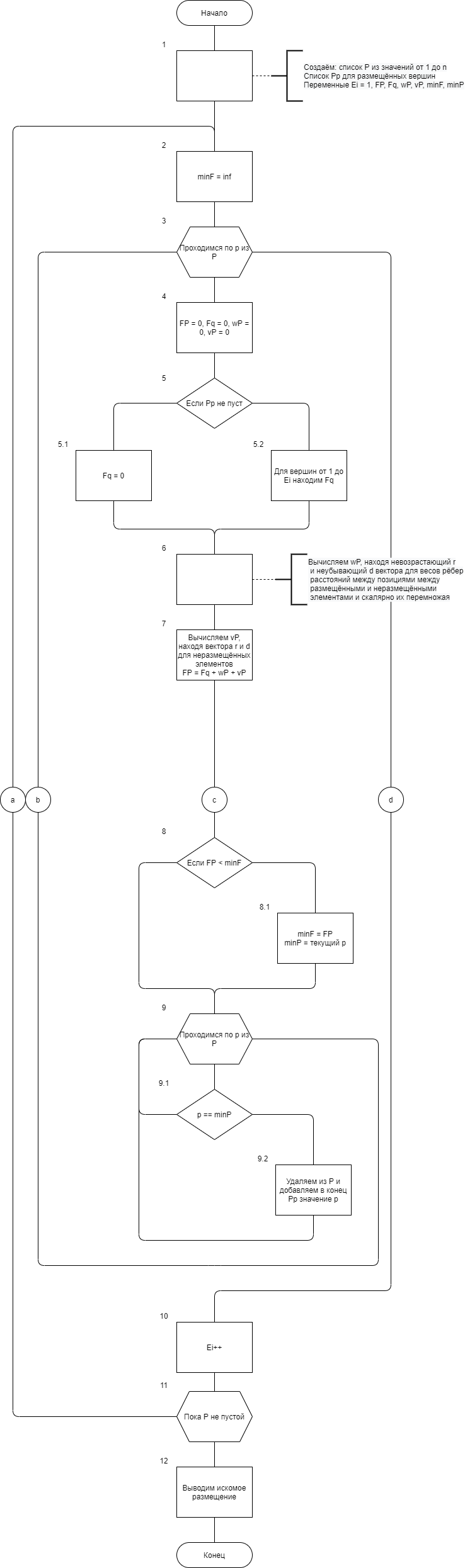


Рис. 1. Алгоритмическая схема, часть 1

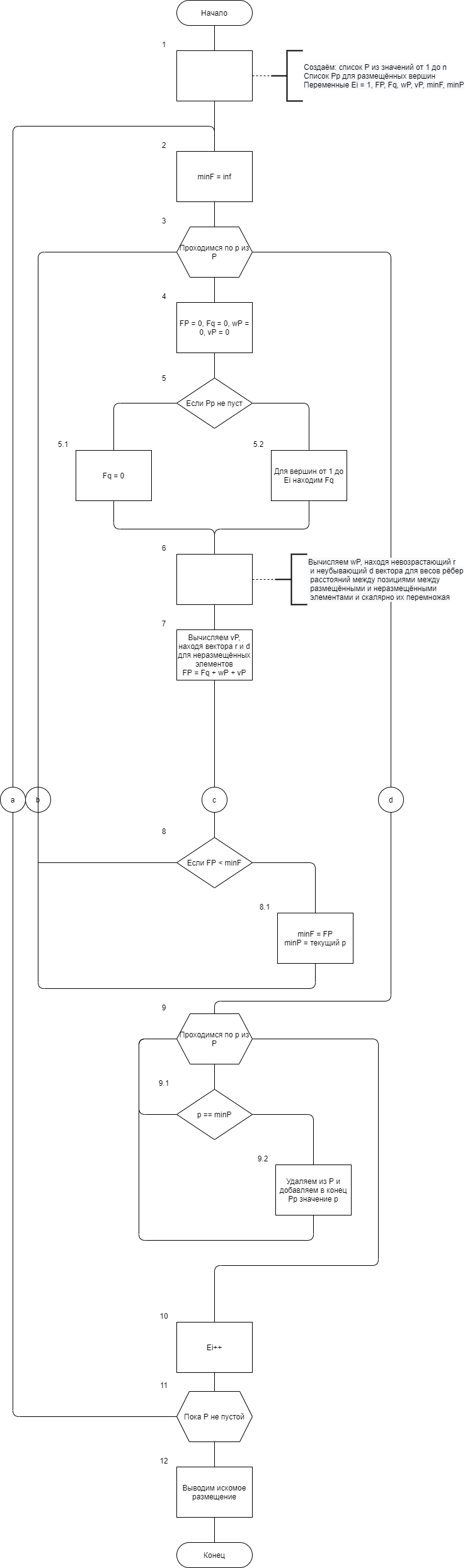


Рис. 2. Алгоритмическая схема, часть 2

**Решение задачи на контрольном примере**

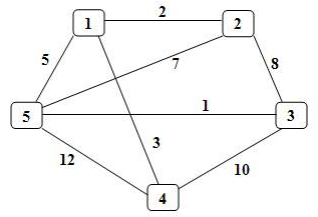


Рис. 3. Граф для решения задачи

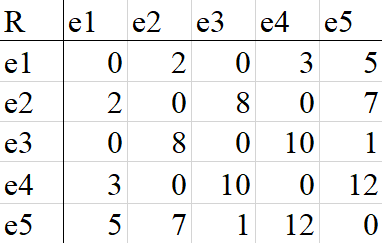


Рис. 4. Матрица смежности

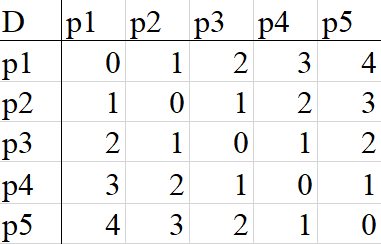


Рис. 5. Матрица расстояний

Шаг1. Инициализация, рис. 6.

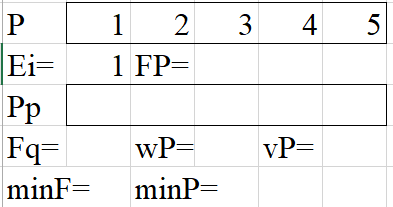
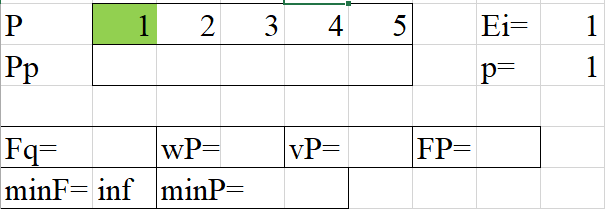


Рис. 6. Инициализация

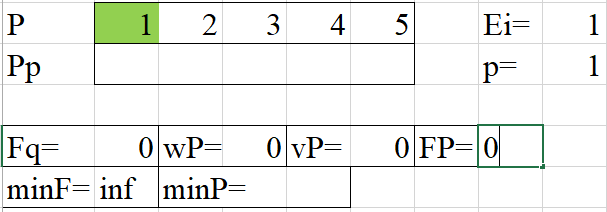
Шаг 2. Задаём минимальное значение большим числом.



Шаг 3. “Устанавливаем” первую вершину в гнездо p1.

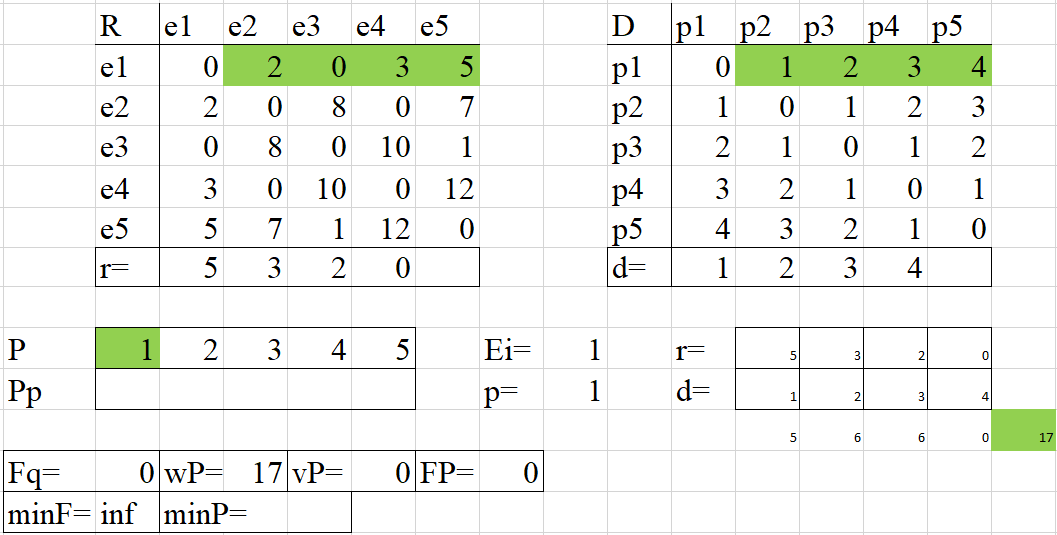


Шаг 4. Обнуляем переменные перед работой тела цикла.

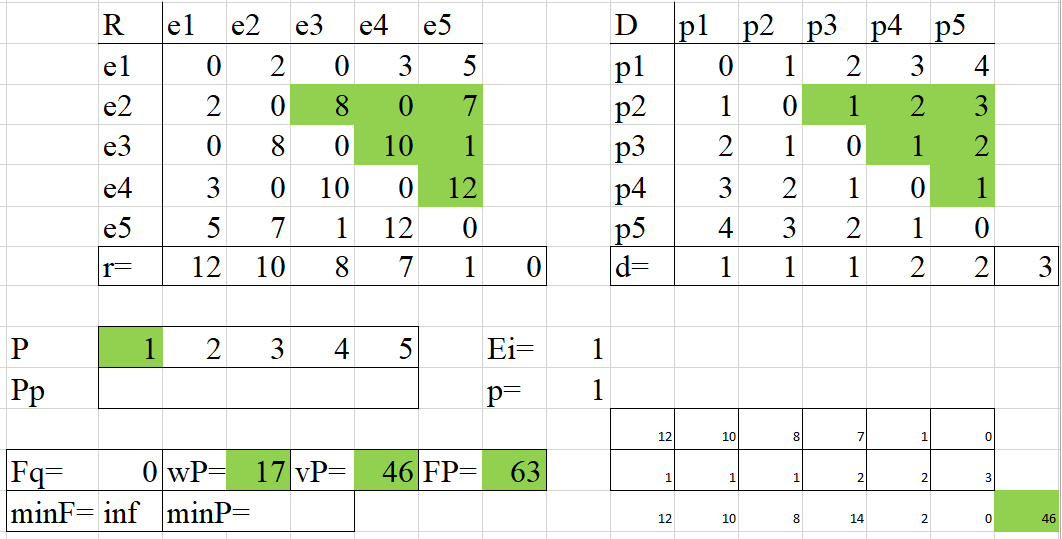


Шаг 5. Рр пока что пуст, поэтому Fq оставляем равным нулю.

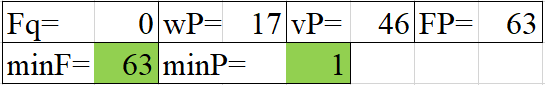
Шаг 6. Так как вершина у нас только одна, находим единственный вектор r (расстояния от первой вершины до всех остальных) и единственный же вектор d, отсортированные. После мы их скалярно перемножили, получив в итоге 17.



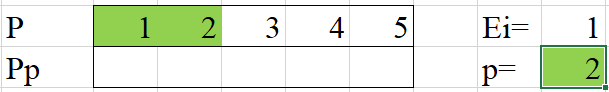
Шаг 7. Зачеркнув строки и столбцы, соответствующие первой вершине и первому гнезду, получаем вектора и их скалярное произведение – vP = 46. Далее сложим полученные Fq, wP, vP, чтобы получить нижнюю оценку FP.



Шаг 8. FP=63 – первое полученное значение, поэтому мы сразу записываем его в minF, а так же запишем, что значение мы нашли при установлении вершины в гнездо p1.

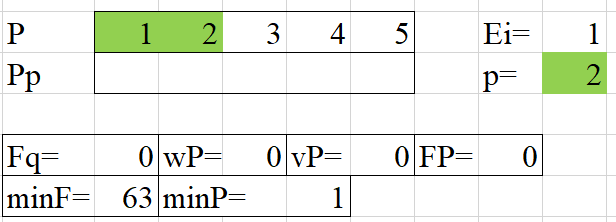


Шаг 9 (3). Возвращаемся к шагу 3, продолжаем брать значения из P, следующее – 2. Вершина всё ещё первая.

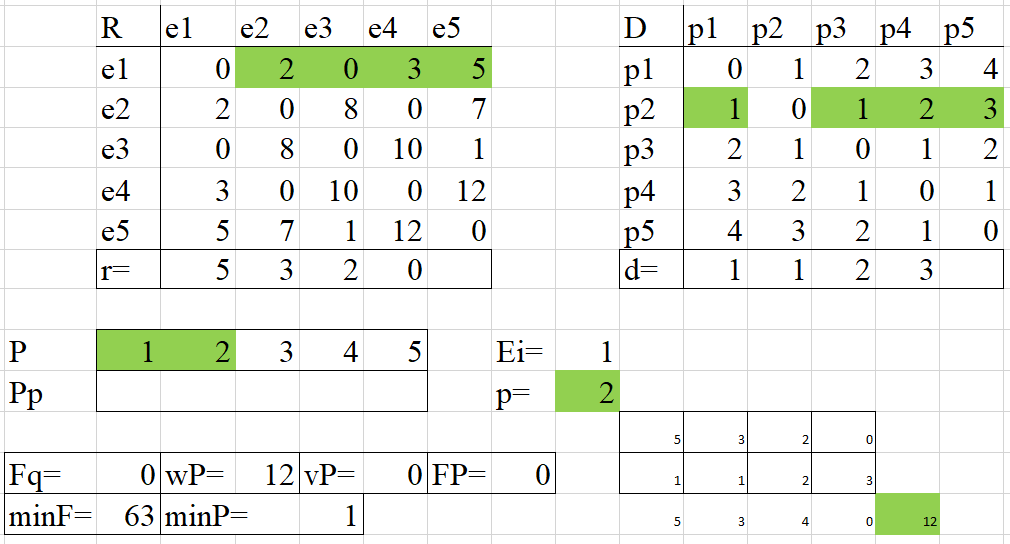


Шаг 10 (4). Обнулили значения.

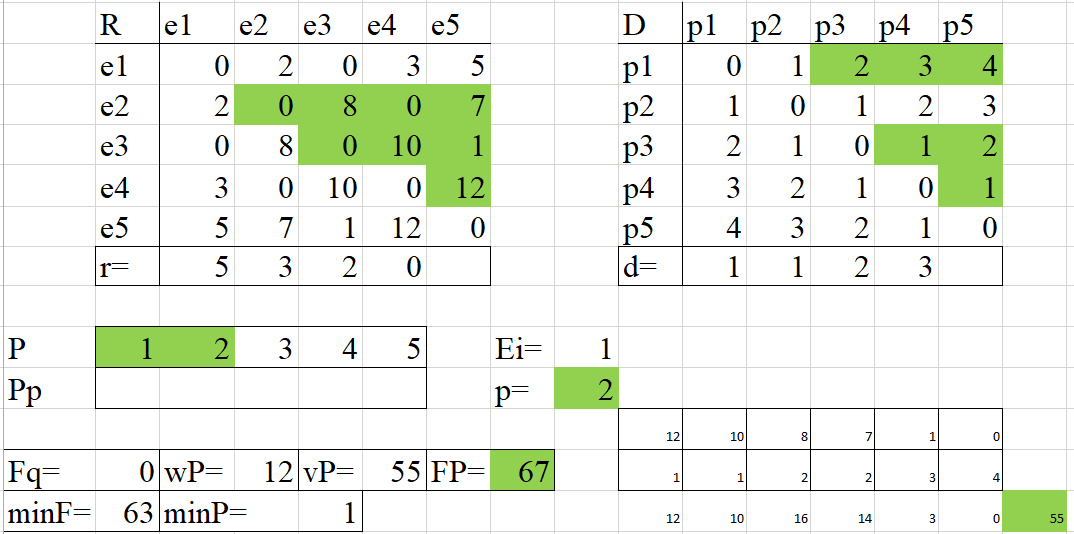
Шаг 11 (5). Рр пока что пуст, поэтому Fq оставляем равным нулю.



Шаг 12 (6). Находим единственный вектор r (расстояния от первой вершины до всех остальных) и единственный же вектор d (расстояние уже от второго гнезда до остальных), отсортированные. После мы их скалярно перемножили, получив в итоге 12.



Шаг 13 (7). Зачеркнув строки и столбцы, соответствующие первой вершине и второму гнезду, получаем вектора и их скалярное произведение – vP = 55. Далее сложим полученные Fq, wP, vP, чтобы получить нижнюю оценку FP.

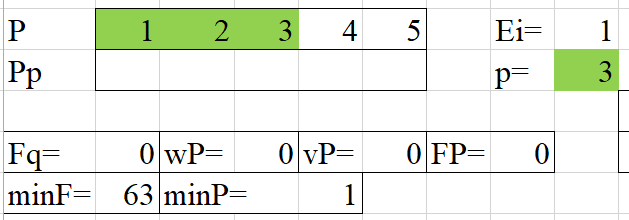


Шаг 14 (8). Так как 67 больше, чем 63, то мы оставляем прошлые значения неизменными.

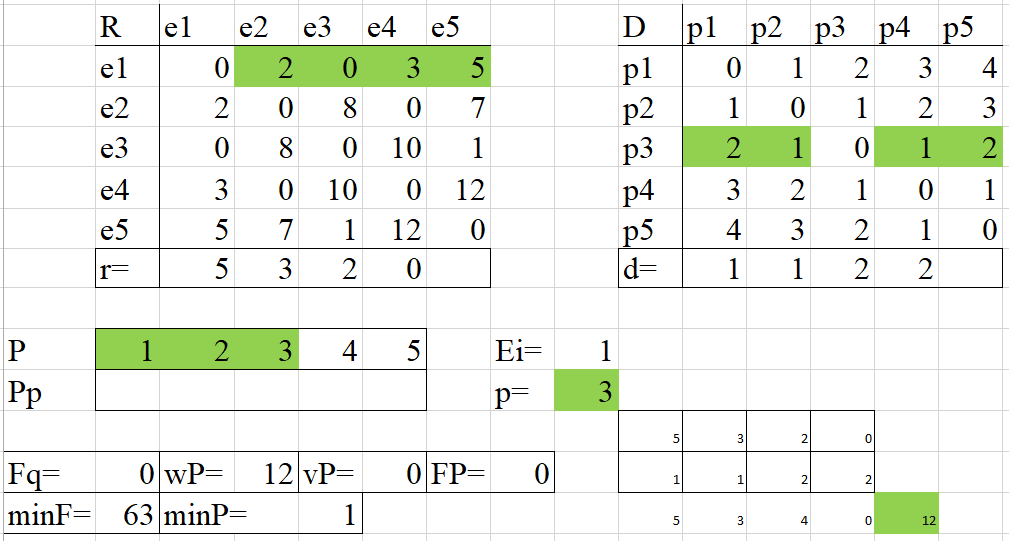
Шаг 15 (3). Возвращаемся к шагу 3, продолжаем брать значения из P, следующее – 3. Вершина всё ещё первая.

Шаг 16 (4). Обнулили значения.

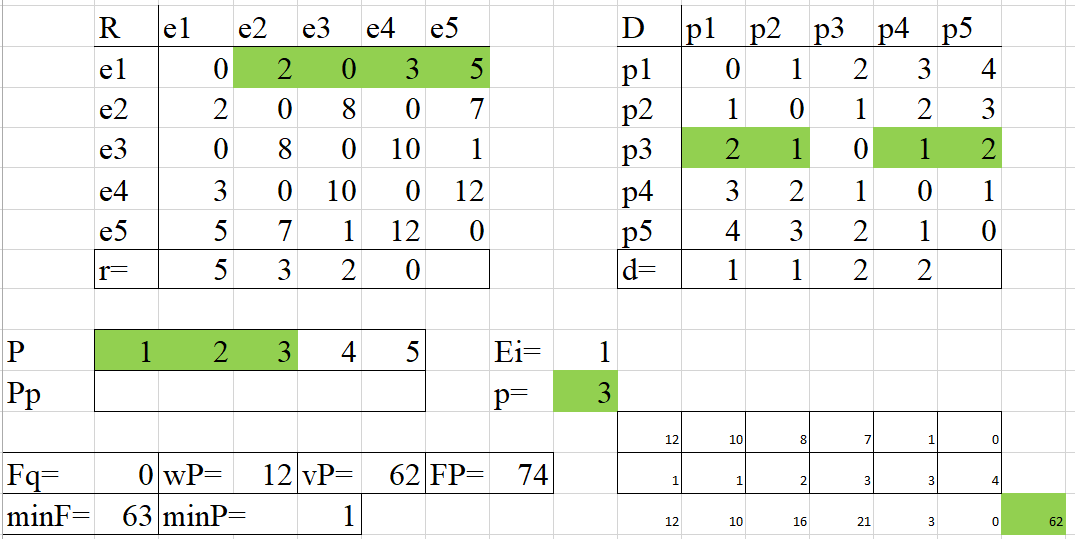
Шаг 17 (5). Рр пока что пуст, поэтому Fq оставляем равным нулю.



Шаг 18 (6). Находим единственный вектор r (расстояния от первой вершины до всех остальных) и единственный же вектор d (расстояние от третьего гнезда до остальных), отсортированные. После мы их скалярно перемножили, получив в итоге 12.



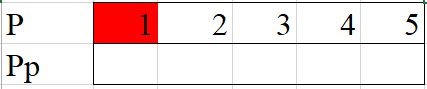
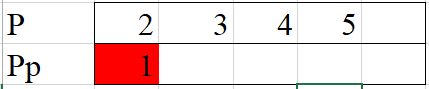
Шаг 19 (7). Зачеркнув строки и столбцы, соответствующие первой вершине и третьему гнезду, получаем вектора и их скалярное произведение – vP = 62. Далее сложим полученные Fq, wP, vP, чтобы получить нижнюю оценку FP.



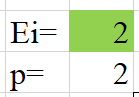
Шаг 20 (8). Так как 74 больше, чем 63, то мы оставляем прошлые значения неизменными.

Примечание: Позиции 4 и 5 симметричны позициям 2 и 1 соответственно, следовательно значение нижней оценки получится абсолютно таким же, и эти шаги мы опустим, сразу перейдя к 9-му шагу основного алгоритма и продолжив уже с позицией номер два.

Шаг 33 (9). minP указывает на гнездо 1, гнездо с минимальной нижней границей. Есть ещё симметричное пятая, но мы по порядку возьмём гнездо 1. Итого, первая вершина располагается на позиции 1.

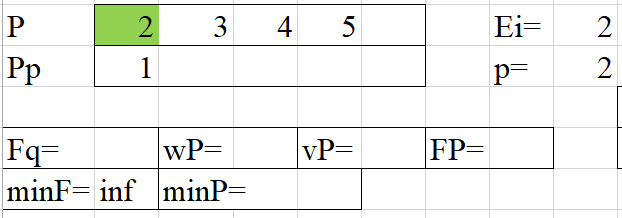
Шаг 34 (10). Перешли к вершине 2.



Шаг 35 (11). Р не пуст, идём по циклу дальше.

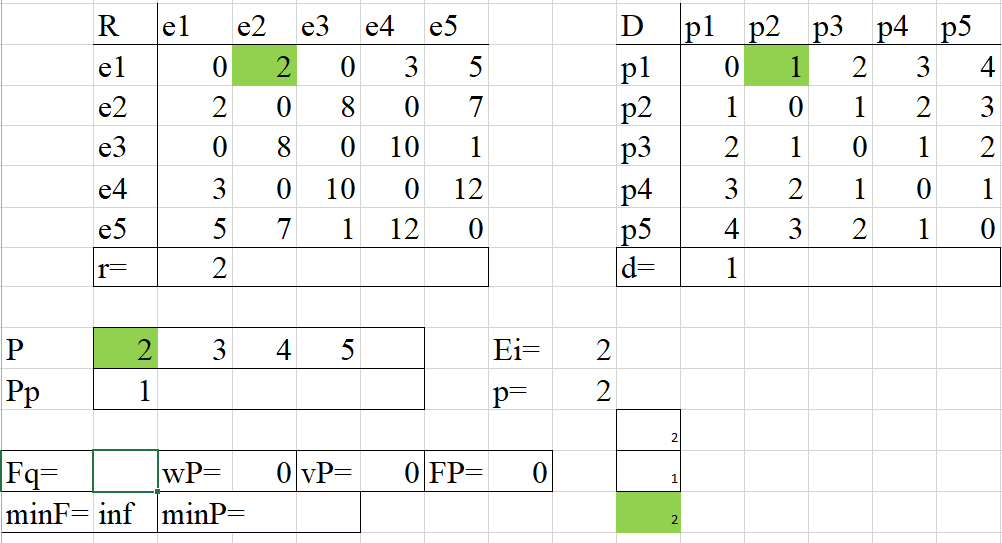
Шаг 36 (2). Обнулим minF для следующего уровня ветвления.

Шаг 37 (3). В Р остались гнёзда 2, 3, 4, 5. Берём гнездо 2.

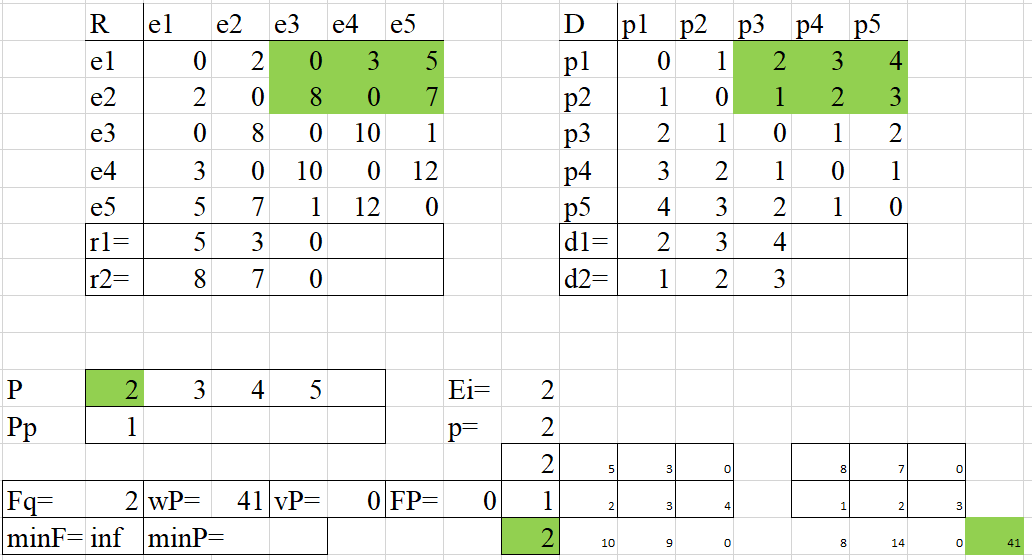


Шаг 38 (4). Обнулим значения.

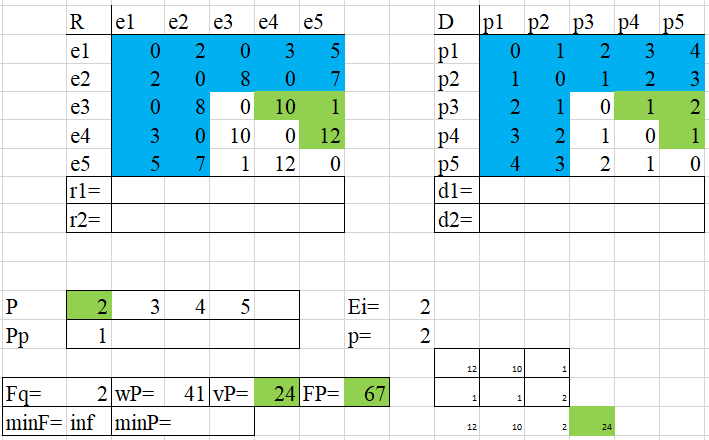
Шаг 39 (5). Так как Рр уже не пуст, будем высчитывать Fq. У нас 2 вершины в размещении, первая и вторая, и вес ребра между ними – 2. Расстояние между позициями – 1. Итого – 2.



Шаг 40 (6). Находим уже два вектора r (расстояния от первой и второй вершин до всех остальных) и два вектора d (расстояние от первого и второго гнёзд до остальных), отсортированные. После мы их скалярно перемножили, получив в итоге 41.



Шаг 41 (7). Зачеркнув строки и столбцы, соответствующие первой, второй вершине и первому, второму гнезду, получаем вектора и их скалярное произведение – vP = 24. Далее сложим полученные Fq, wP, vP, чтобы получить нижнюю оценку FP.

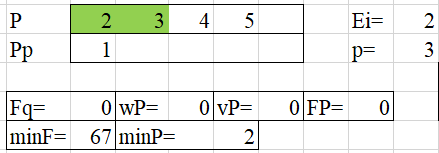


Шаг 42 (8). 67 – первое значение, оно меньше максимума, его и берём. Также указываем позицию, в которой мы нашли эту оценку.

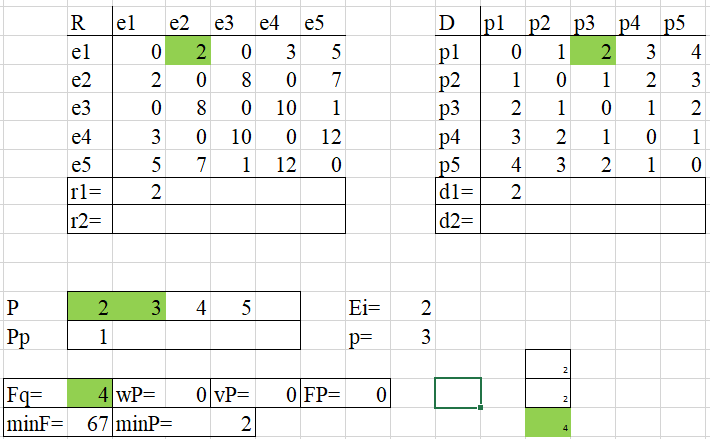


Шаг 43 (3). Возвращаемся к шагу 3, продолжаем брать значения из P, следующее – 3. Вершины – первая и вторая.

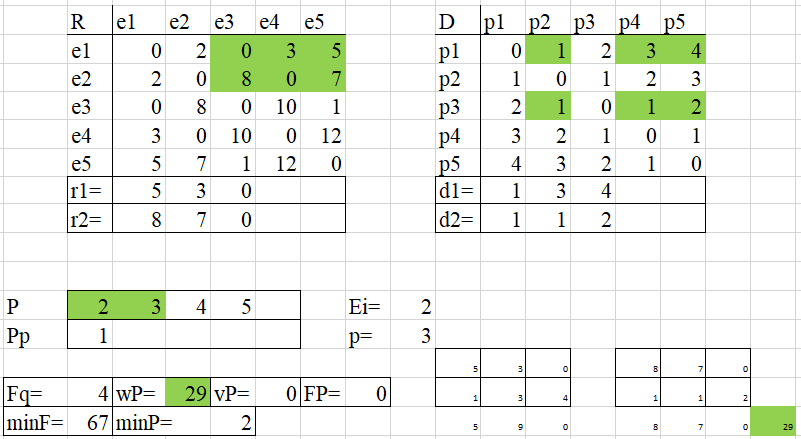
Шаг 44 (4). Обнуляем значения.



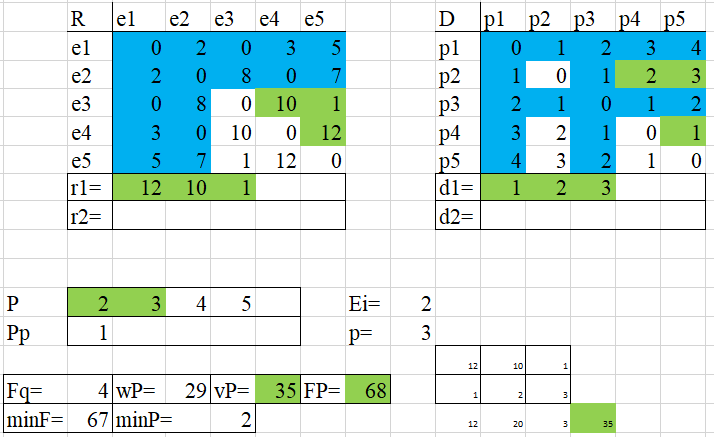
Шаг 45 (5). Так как Рр уже не пуст, будем высчитывать Fq. У нас 2 вершины в размещении, первая и вторая, и вес ребра между ними – 2. Расстояние между позициями – уже 2. Итого – 4.



Шаг 46 (6). Находим два вектора r (расстояния от первой и второй вершин до всех остальных) и два вектора d (расстояние от первого и третьего гнёзд до остальных), отсортированные. После мы их скалярно перемножили, получив в итоге 29.



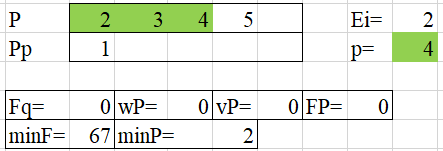
Шаг 47 (7). Зачеркнув строки и столбцы, соответствующие первой, второй вершине и первому, третьему гнезду, получаем вектора и их скалярное произведение – vP = 35. Далее сложим полученные Fq, wP, vP, чтобы получить нижнюю оценку FP.



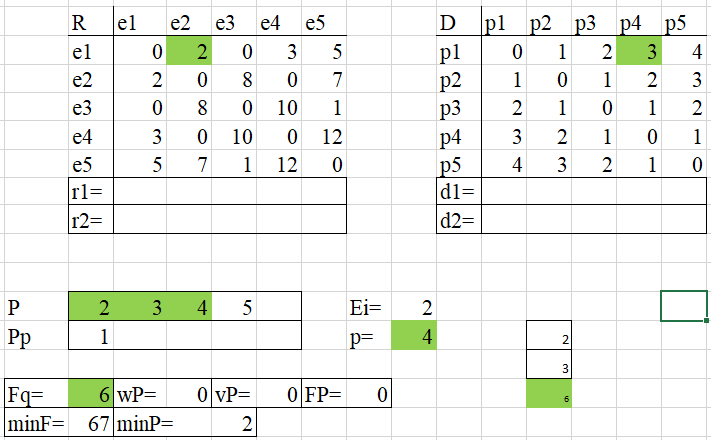
Шаг 48 (8). Так как 68 больше, чем 67, то мы оставляем прошлые значения неизменными.

Шаг 49 (3). Возвращаемся к шагу 3, продолжаем брать значения из P, следующее – 4. Вершины – первая и вторая.

Шаг 50 (4). Обнуляем значения.



Шаг 51 (5). Так как Рр уже не пуст, будем высчитывать Fq. У нас 2 вершины в размещении, первая и вторая, и вес ребра между ними – 2. Расстояние между позициями – уже 3. Итого – 6.



Шаг 52 (6). Находим два вектора r (расстояния от первой и второй вершин до всех остальных) и два вектора d (расстояние от первого и четвёртого гнёзд до остальных), отсортированные. После мы их скалярно перемножили, получив в итоге 29.

Шаг 53

Шаг 54

Шаг 55

Шаг 56

Шаг 57

Шаг 58

Шаг 59

Шаг 60

Шаг 61

Шаг 62

Шаг 63

Шаг 64

Шаг 65

Шаг 66

Шаг 67

Шаг 68

Шаг 69

Шаг 70