ФПМИ, 3 курс, 9а группа

Крагель Алина Олеговна

ИСО

Исаченко Александр Николаевич

Лабораторная работа №5

1. Алгоритм Флойда применяется для решения задачи о нахождении для каждой пары вершин графа кратчайший путь. Ограничений на длины дуг не накладывается. Алгоритм обнаруживает контур отрицательной длины в графе.

По окончанию применения алгоритма если k+1 = n и то матрица даёт кратчайшие расстояния между парами вершин.

Для условия задания 1 строим матрицы по матрицам :

Таким образом, после проведения 5 итераций алгоритма Флойда, получили матрицу с неотрицательными диагональными элементами, что дает нам право назвать матрицу корректной матрицей кратчайших расстояний между вершинами данного условием графа.

1. Алгоритм Форда-Фалкерсонадля нахождения максимального потока начинает свою работу с произвольного начального потока в сети. Например, нулевого потока. Алгоритм на каждой итерации состоит из двух этапов: расстановка меток и увеличение потока. Изменение потоков на величину по дугам повторяется до тех пор, пока не будет достигнута вершина s. Стираем у вершин все метки и возвращаемся к этапу 1 с новым увеличенным потоком.

*X3*

*X2*

*X1*

*s*

*t*

4

4

4

3

1

3

2

2

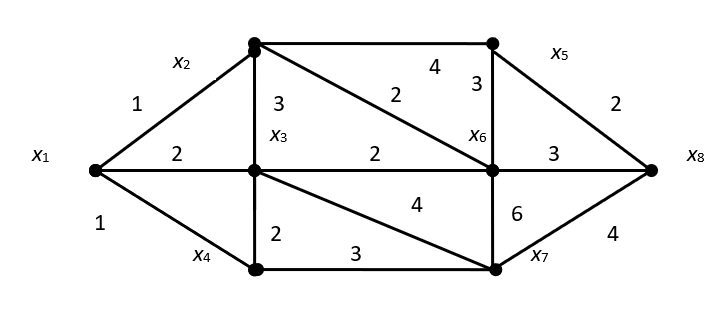
4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | *s* |  |  |  | *t* |  |
| 1. |  |  |  |  |  | 0 + 3 |
| 2. |  |  |  |  |  | 3 + 1 |
| 3. |  |  |  |  |  | 4 + 2 |
| 4. |  |  |  |  |  | 6 |

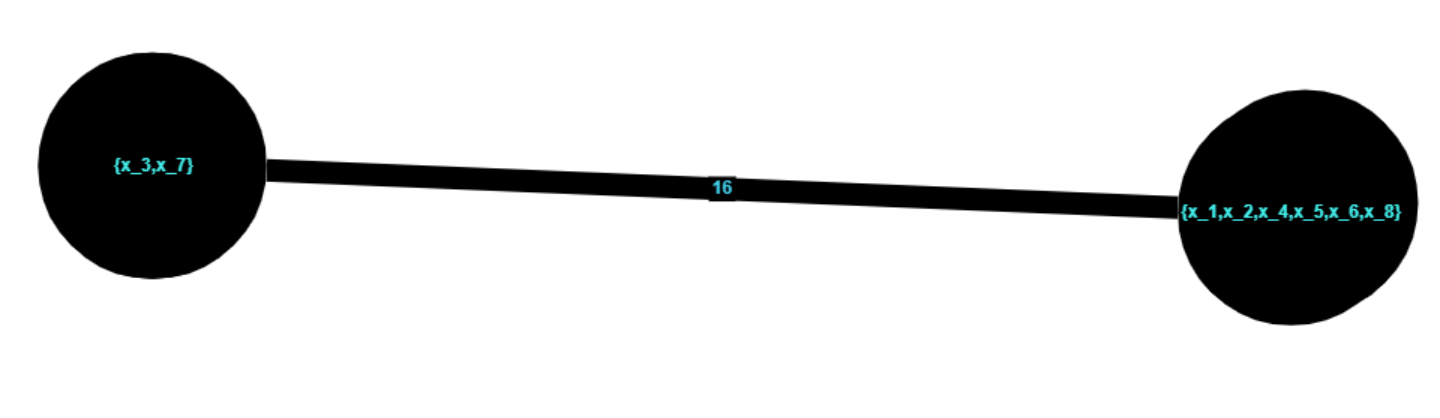
Таким образом из таблицы решений:

* максимальный поток ;
* минимальный срез

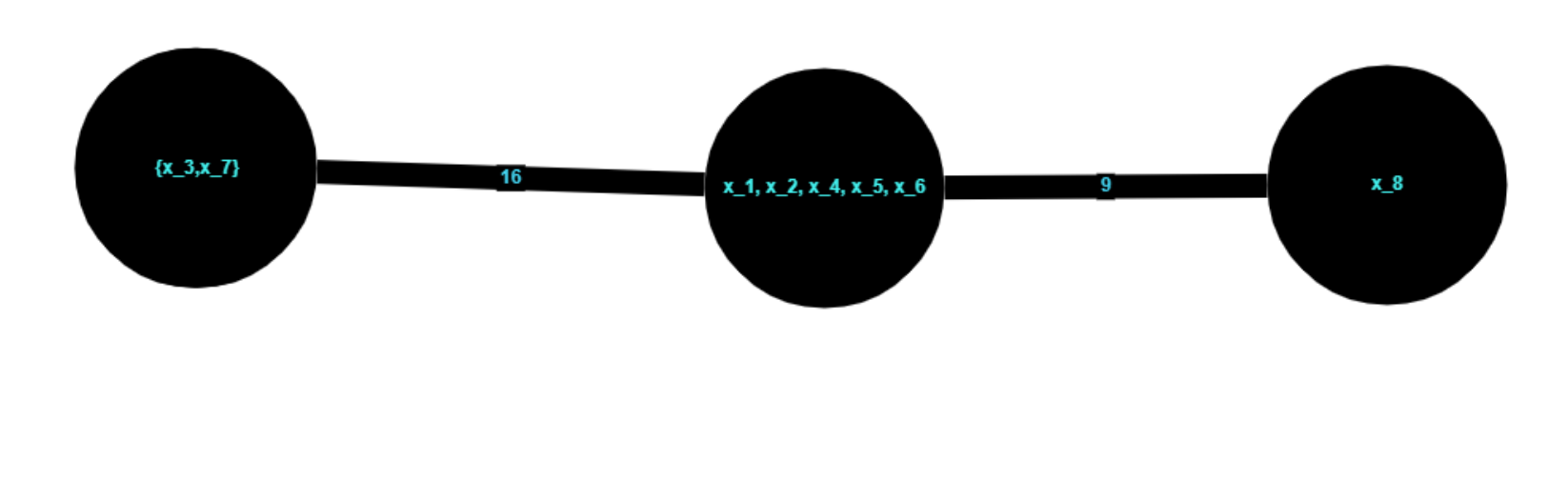
1. Алгоритм Гомори-Ху является более эффективным, ибо в нем задача о максимальном потоке решается *n*-1 раз, нежели при решении данной задачи алгоритмом Форда-Фалкерсона. Идея алгоритма Гомори-Ху состоит в итеративном построении максимального остовного дерева . Если требуется определить величину максимального потока между двумя произвольными узлами, надо в дереве найти путь, соединяющий эти два узла, и выбрать в этом пути дугу с минимальным весом. Вес этой дуги равен величине максимального потока между рассматриваемыми узлами.

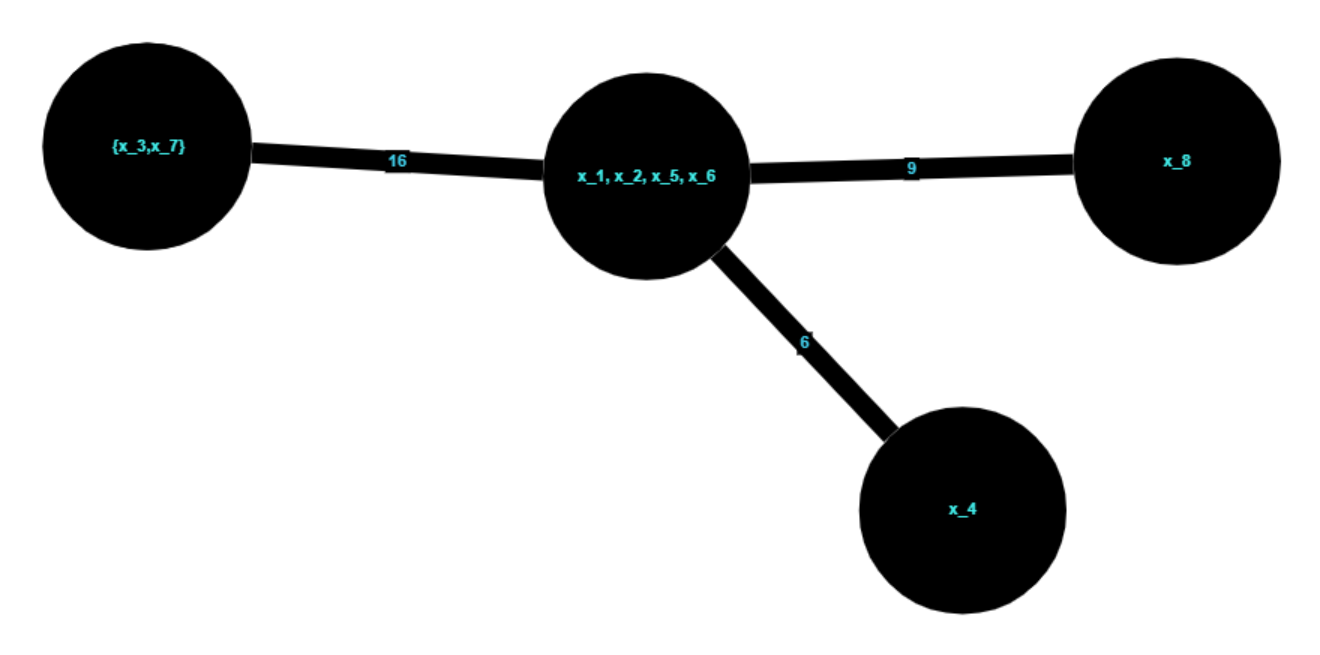


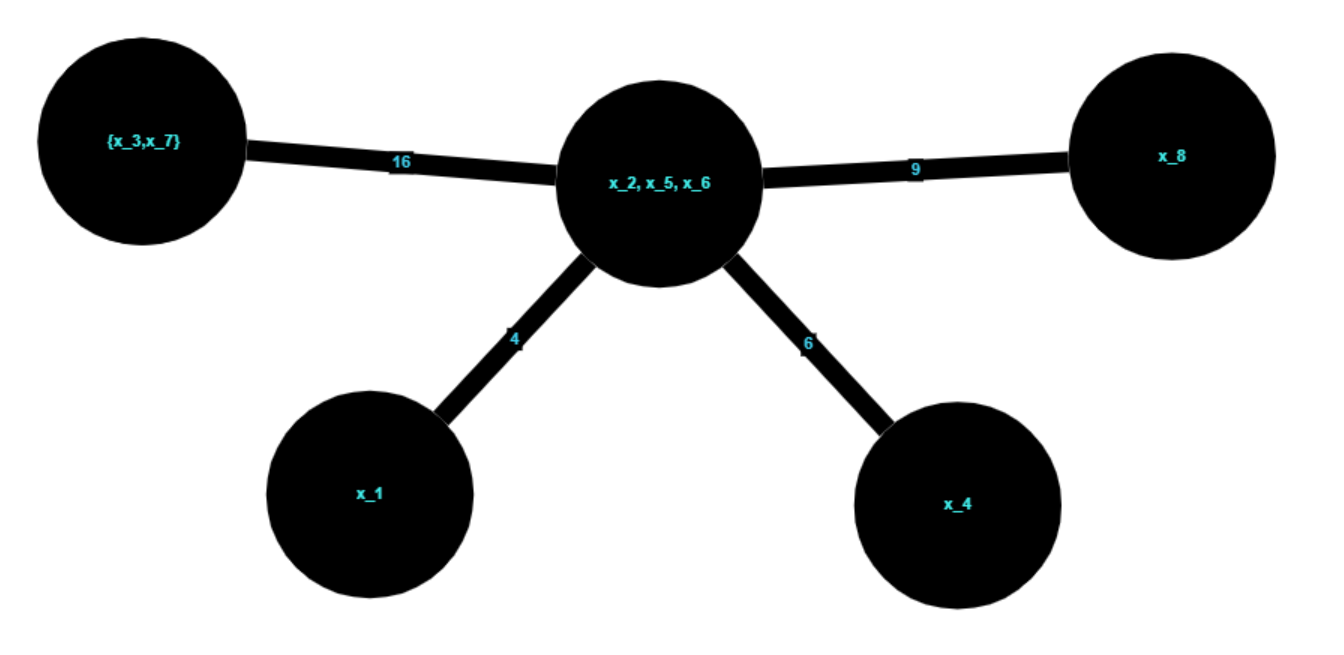
*Итерация 1.* Возьмём *s = x7, t = x6*. Минимальный разрез, отделяющий *x7* от *x6*, есть . Его пропускная способность равна . По минимальному разрезу получим, что дерево на первой итерации состоит из двух множеств вершин: , и единственного ребра с весом равным 16.



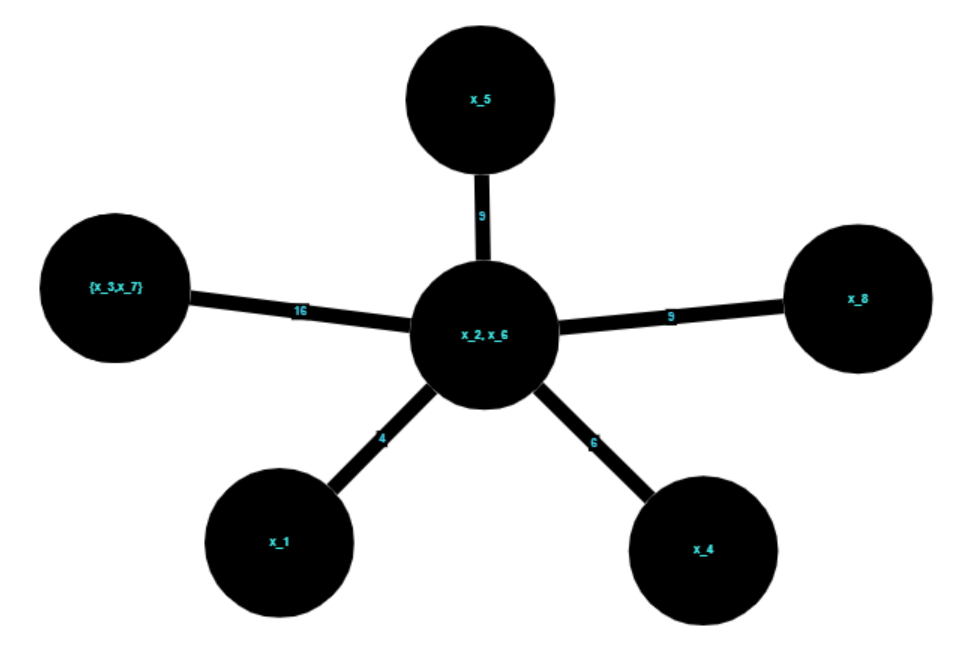
*Итерация 2. s = x8, t = x5*. Минимальный разрез, отделяющий *x8* от *x5*, есть . Его пропускная способность равна . По минимальному разрезу получим, что дерево на второй итерации состоит из трех множеств вершин: , и добавленного ребра с весом равным 9.



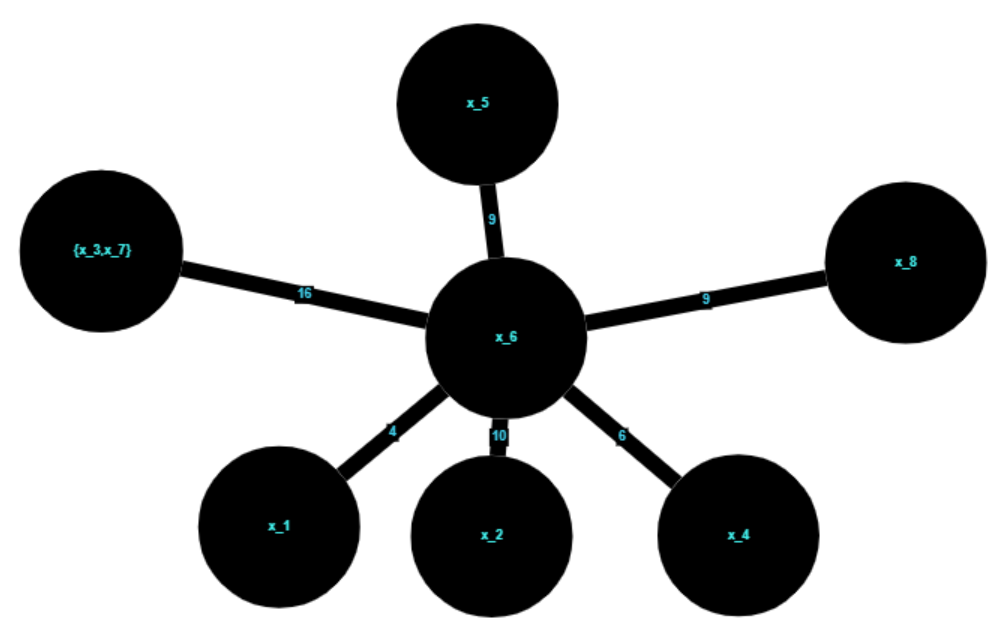
*Итерация 3. s=x4, t=x5*. Минимальный разрез, отделяющий *x4* от *x5*, есть . Его пропускная способность равна 6. дерево на второй итерации состоит из трех множеств вершин: , и добавленного ребра с весом равным 6.

*Итерация 4. s=x1, t=x2*. Минимальный разрез: , пропускная способность равна 4. Добавим новую вершину и дугу.

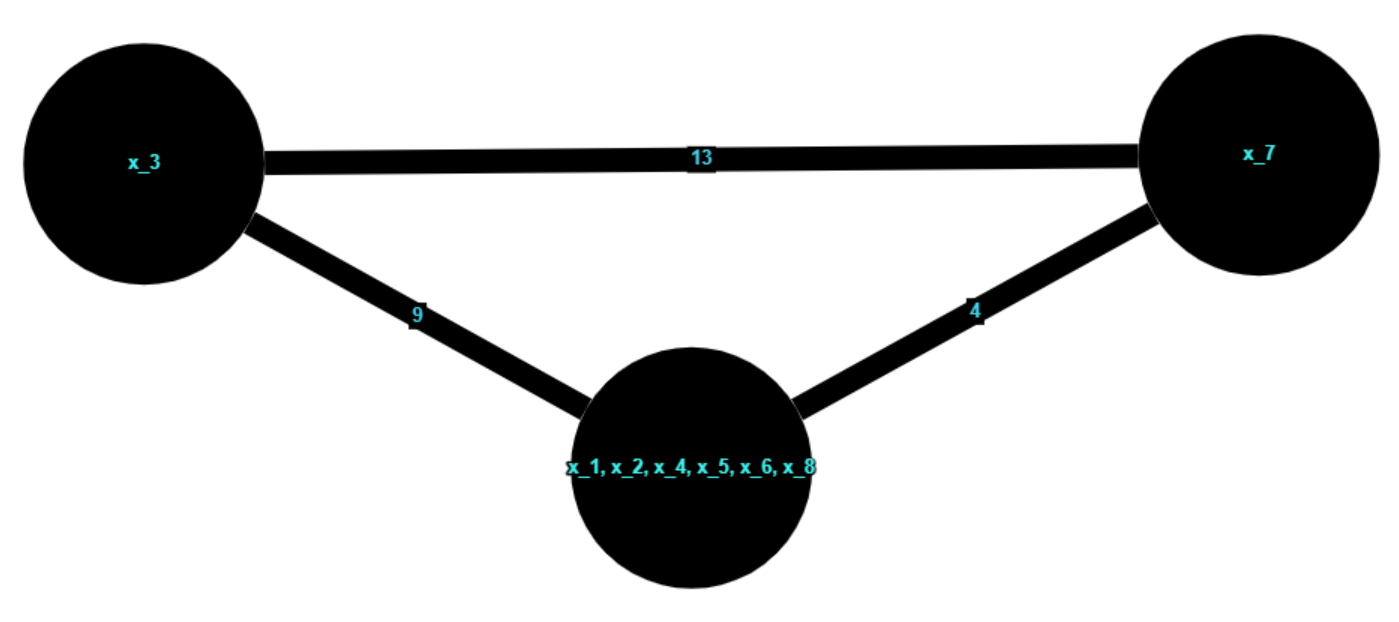
*Итерация 5. s=x5, t=x2*. Минимальный разрез: , пропускная способность равна 9.



*Итерация 6. s=x2, t=x6*. Минимальный разрез: , пропускная способность равна 10.



*Итерация 7.* Исходный граф:



*s=x3, t=x7*. Минимальный разрез . Новая вершина {x3}, ребро веса 13 от изменённой вершины {x7}:

