Спецификация

**Тема:** Алгоритм Флойда- Уоршелла

**Группа:** 5303

**Студенты:** Матвеева А.И.

Нигай А.С.

Табунникова Н.Р.

**Теория**

Алгоритм Флойда-Уоршелла — алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного [графа](http://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) без циклов с отрицательными весами с использованием метода динамического программирования. Алгоритм легко модифицировать таким образом, чтобы он возвращал не только длину кратчайшего пути, но и сам путь. Для этого достаточно завести дополнительный массив next, в котором будет храниться номер вершины, в которую надо пойти следующей, чтобы дойти из u в v по кратчайшему пути.

**Постановка задачи**

Дан взвешенный ориентированный [граф](https://neerc.ifmo.ru/wiki/index.php?title=%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5_%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F:_%D0%B3%D1%80%D0%B0%D1%84,_%D1%80%D0%B5%D0%B1%D1%80%D0%BE,_%D0%B2%D0%B5%D1%80%D1%88%D0%B8%D0%BD%D0%B0,_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D1%8C,_%D0%BF%D0%B5%D1%82%D0%BB%D1%8F,_%D0%BF%D1%83%D1%82%D1%8C,_%D1%86%D0%B8%D0%BA%D0%BB) G(V, E), в котором вершины пронумерованы от 1 до n.

Требуется найти матрицу кратчайших расстояний d, в которой элемент dij  либо равен длине кратчайшего пути из i в j, либо равен 0, если вершина j не достижима из i.

**Алгоритм**

Пусть вершины графа {\displaystyle G=(V,\;E),\;|V|=n}G=(V,E), |V|=n  пронумерованы от 1 до {\displaystyle n}n и введено обозначение   
{\displaystyle d\_{ij}^{k}} для длины кратчайшего пути от {\displaystyle i}i до {\displaystyle j}j, который кроме самих вершин {\displaystyle i,\;j}i, j проходит только через вершины {\displaystyle 1\ldots k}1..k. Очевидно, что {\displaystyle d\_{ij}^{0}} — длина (вес) ребра {\displaystyle (i,\;j)}(i,j), если таковое существует (в противном случае его длина может быть обозначена как {\displaystyle \infty }∞).

Существует два варианта значения {\displaystyle d\_{ij}^{k},\;k\in \mathbb {(} 1,\;\ldots ,\;n)}:

1. Кратчайший путь между {\displaystyle i,\;j}i, j не проходит через вершину {\displaystyle k}k, тогда {\displaystyle d\_{ij}^{k}=d\_{ij}^{k-1}}
2. Существует более короткий путь между {\displaystyle i,\;j}i, j, проходящий через {\displaystyle k}k, тогда он сначала идёт от {\displaystyle i}i до {\displaystyle k}k, а потом от {\displaystyle k}k до {\displaystyle j}i. В этом случае, очевидно, {\displaystyle d\_{ij}^{k}=d\_{ik}^{k-1}+d\_{kj}^{k-1}}

Таким образом, для нахождения значения функции достаточно выбрать минимум из двух обозначенных значений.

Тогда [рекуррентная](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%81%D0%B8%D1%8F) формула для {\displaystyle d\_{ij}^{k}} имеет вид:

{\displaystyle d\_{ij}^{0}} — длина ребра {\displaystyle (i,\;j);}(i, j);

{\displaystyle d\_{ij}^{k}=\min(d\_{ij}^{k-1},\;d\_{ik}^{k-1}+d\_{kj}^{k-1}).}

Алгоритм Флойда-Уоршелла последовательно вычисляет все значения {\displaystyle d\_{ij}^{k},} {\displaystyle \forall i,\;j} для {\displaystyle k}k от 1 до {\displaystyle n}n. Полученные значения {\displaystyle d\_{ij}^{n}} являются длинами кратчайших путей между вершинами {\displaystyle i,\;j.}i, j.

Алгоритм работает за O(n3) времени и использует O(n2) памяти.

**Реализация**

На вход программе подаётся граф, заданный в виде матрицы смежности — двумерного массива d[][] размера n×n, в котором каждый элемент задаёт длину ребра между соответствующими вершинами.

Требуется, чтобы выполнялось d[i][i]=0 для любых i.

**Псевдокод**

На каждом шаге алгоритм генерирует [матрицу](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) {\displaystyle W,}W, .  
{\displaystyle w\_{ij}=d\_{ij}^{n}.}Матрица {\displaystyle W}W содержит длины кратчайших путей между всеми вершинами графа. Перед работой алгоритма матрица {\displaystyle W}W заполняется длинами рёбер графа (или запредельно большим *M*, если ребра нет).

**for** k = 1 **to** n

**for** i = 1 **to** n

**for** j = 1 **to** n

W[i][j] = min(W[i][j], W[i][k] + W[k][j])

**Особые случаи**

**Случай вещественных весов**

Если веса рёбер графа не целочисленные, а вещественные, то следует учитывать погрешности, неизбежно возникающие при работе с типами с плавающей точкой.

Применительно к алгоритму Флойда неприятным спецэффектом этих погрешностей становится то, что найденные алгоритмом расстояния могут уйти сильно в минус из-за накопившихся ошибок. В самом деле, если на первой фазе имела место ошибка ∆, то на второй итерации эта ошибка уже может превратиться в 2∆, на третьей — в 4∆, и так далее.

Чтобы этого не происходило, сравнения в алгоритме Флойда следует делать с учётом погрешности:

**if** (d[i][k] + d[k][j] < d[i][j] - EPS)

d[i][j] = d[i][k] + d[k][j];

**Случай отрицательных циклов**

Если в графе есть циклы отрицательного веса, то формально алгоритм Флойда-Уоршелла неприменим к такому графу.

На самом же деле, для тех пар вершин i и j, между которыми нельзя зайти в цикл отрицательного вес, алгоритм отработает корректно.

Для тех же пар вершин, ответа для которых не существует (по причине наличия отрицательного цикла на пути между ними), алгоритм Флойда найдёт в качестве ответа какое-то число (возможно, сильно отрицательное, но не обязательно). Тем не менее, можно улучшить алгоритм Флойда, чтобы он аккуратно обрабатывал такие пары вершин и выводил для них, например, -.

Для этого можно сделать, например, следующий критерий "не существования пути". Итак, пусть на данном графе отработал обычный алгоритм Флойда. Тогда между вершинами i и j не существует кратчайшего пути тогда и только тогда, когда найдётся такая вершина t, достижимая из i и из которой достижима j, для которой выполняется d[t][t]<0.

Кроме того, при использовании алгоритма Флойда для графов с отрицательными циклами следует помнить, что возникающие в процессе работы расстояния могут сильно уходить в минус, экспоненциально с каждой фазой. Поэтому следует принять меры против целочисленного переполнения, ограничив все расстояния снизу какой-нибудь величиной (например, -INF).

# Требования к программе

Программа должна выполнять следующие действия:

* По введенной матрице смежности/ матрице инцидентности/ списку инцидентности отображать граф, работу алгоритма Флойда – Уоршелла;
* Генерировать новый граф;
* Иметь и при надобности выводить справку для пользователя.

**Эскиз интерфейса**

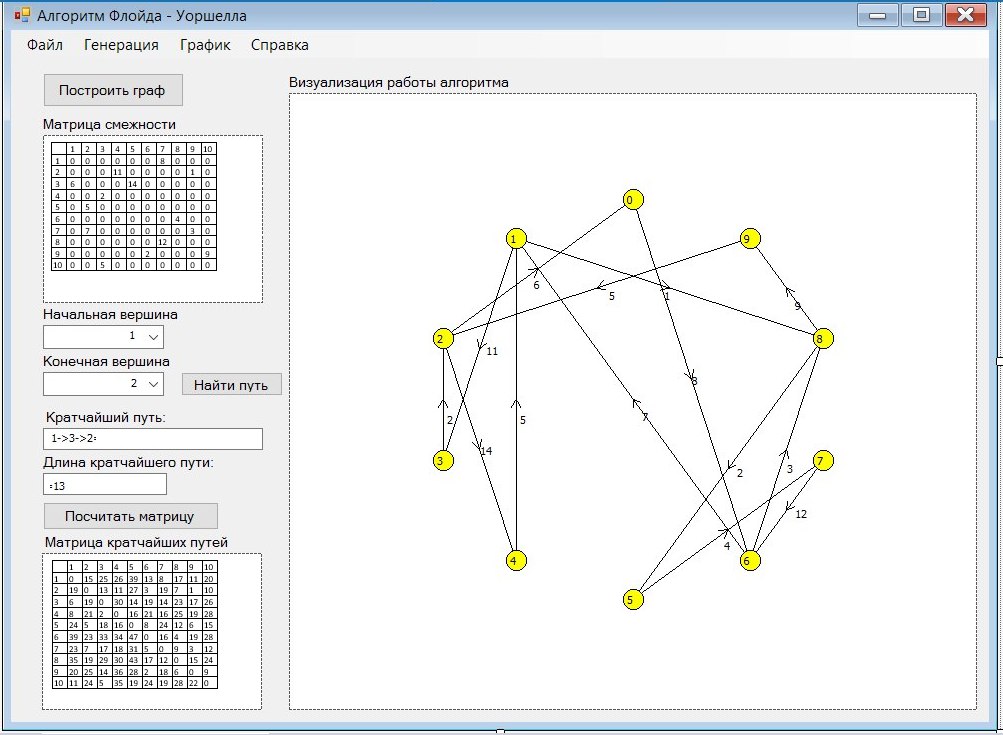


Рис. 1

**Работа меню:**

**Файл** – взять матрицу смежности из файла. В файле матрица смежности указывается как последовательность чисел, где количество вершин определяется автоматически. 0 означает, что ребро отсутствует, а любое другое целое число является весом ребра. Файл выбирается пользователем.

**Генерация** – сгенерировать матрицу смежности по заданному количеству вершин со случайным (в диапазоне, зависящем от количества вершин) количеством ребер.

**Справка** – получить справку (руководство пользователя). Справка открывается в отдельном окне.

**Работа программы:**

Программа может выполнять следующие действия:

* построить граф по матрице смежности;
* посчитать матрицу кратчайших путей для заданного графа;
* найти кратчайший путь между двумя вершинами заданного графа;

Для того чтобы построить граф, нужно ввести матрицу смежности в соответствующее поле и нажать кнопку **«Построить граф»**. Граф построится в поле для визуализации работы алгоритма. Граф рисуется по матрице смежности. Пользователь по желанию может ввести вместо матрицы список инцидентности. Вид представления графа пользователь должен указать в специальной форме. При желании пользователь может взять матрицу смежности из файла или сгенерировать ее. Тогда граф построится автоматически.

Для того чтобы посчитать матрицу кратчайших путей для заданного графа, нужно нажать кнопку **«Посчитать матрицу»**. Матрица кратчайших путей появится в соответствующем поле. При работе алгоритма должны подсвечиваться проверяемые на данной итерации вершины и пути и меняться матрица кратчайших путей в удобном для пользователя виде. Последние изменения в матрице кратчайших путей будут выделяться жирным шрифтом.

Для того чтобы найти кратчайший путь между двумя вершинами заданного графа, нужно нажать кнопку **«Найти путь»** и выбрать начальную и конечную вершины на графе. Кратчайший путь должен подсветиться на графе и в матрице кратчайших путей, а его длина появится в соответствующем поле.

# Формат входных данных:

Для получения корректного результата программе нужна матрица смежности / матрица инцидентности / список инцидентности в виде последовательности целых чисел.

# План разработки:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Создание прототипа, 26 июня | 1-ая версия, 28 июня | 2-ая (конечная) версия, 30 июня |
| Нигай А.С. | Реализация визуализации графа. | Визуализация работы алгоритма Флойда – Уоршелла, разработка метода генерации графа. | Разработка метода нахождения циклов отрицательного веса. |
| Табунникова Н.Р. | Разработка системы ввода/вывода. | Разработка механизмов обеспечивающих корректность вводимых данных. | Написание справки |
| Матвеева А.И. | Реализация методов поиска всех кратчайших путей графа алгоритмом Флойда-Уоршелла, поиска кратчайшего пути между двумя вершинами. | Разработка метода преобразования списка инцидентности в матрицу смежности. |