Тема:
Алгоритм Флойда

Сергей Витальевич Рыбин

svrybin@etu.ru

СПбГЭТУ «ЛЭТИ», кафедра «Алгоритмической математики»

29 июня 2023 г.

Основные понятия

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.
2. **Идея алгоритма**. Проводится расширение множества промежуточных вершин для путей, по которым ищутся кратчайшие расстояния.

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.
2. **Идея алгоритма**. Проводится расширение множества промежуточных вершин для путей, по которым ищутся кратчайшие расстояния.
3. Алгоритм начинает работу с матрицы весов (матрицы пометок $d(i, j)$), рассматриваемых как длины кратчайших путей от вершины v_i к вершине v_j, не имеющих ни одной промежуточной вершины, т. е. вес ребра (дуги) (v_i, v_j).

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.
2. **Идея алгоритма**. Проводится расширение множества промежуточных вершин для путей, по которым ищутся кратчайшие расстояния.
3. Алгоритм начинает работу с матрицы весов (матрицы пометок $d(i, j)$), рассматриваемых как длины кратчайших путей от вершины v_i к вершине v_j, не имеющих ни одной промежуточной вершины, т. е. вес ребра (дуги) (v_i, v_j).
4. Далее поочередно каждую вершину представляют в роли промежуточной и через нее осуществляют модификацию пометки, т. е. на каждом шаге одну из вершин добавляют к множеству промежуточных вершин.

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.
2. **Идея алгоритма**. Проводится расширение множества промежуточных вершин для путей, по которым ищутся кратчайшие расстояния.
3. Алгоритм начинает работу с матрицы весов (матрицы пометок $d(i, j)$), рассматриваемых как длины кратчайших путей от вершины v_i к вершине v_j, не имеющих ни одной промежуточной вершины, т. е. вес ребра (дуги) (v_i, v_j).
4. Далее поочередно каждую вершину представляют в роли промежуточной и через нее осуществляют модификацию пометки, т. е. на каждом шаге одну из вершин добавляют к множеству промежуточных вершин.
5. Аналогично рассмотренному ранее алгоритму Дейкстры, легко модифицировать таким образом, чтобы он возвращал не только длину кратчайшего пути, но и сам путь.

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.
2. **Идея алгоритма**. Проводится расширение множества промежуточных вершин для путей, по которым ищутся кратчайшие расстояния.
3. Алгоритм начинает работу с матрицы весов (матрицы пометок $d(i, j)$), рассматриваемых как длины кратчайших путей от вершины v_i к вершине v_j, не имеющих ни одной промежуточной вершины, т. е. вес ребра (дуги) (v_i, v_j).
4. Далее поочередно каждую вершину представляют в роли промежуточной и через нее осуществляют модификацию пометки, т. е. на каждом шаге одну из вершин добавляют к множеству промежуточных вершин.
5. Аналогично рассмотренному ранее алгоритму Дейкстры, легко модифицировать таким образом, чтобы он возвращал не только длину кратчайшего пути, но и сам путь.
6. Для этого, сохранив преемственность, введем дополнительную матрицу G, в которой для каждой пары вершин u и v будем хранить номер первой промежуточной вершины кратчайшего пути из u в v.

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.
2. **Идея алгоритма**. Проводится расширение множества промежуточных вершин для путей, по которым ищутся кратчайшие расстояния.
3. Алгоритм начинает работу с матрицы весов (матрицы пометок $d(i, j)$), рассматриваемых как длины кратчайших путей от вершины v_i к вершине v_j, не имеющих ни одной промежуточной вершины, т. е. вес ребра (дуги) (v_i, v_j).
4. Далее поочередно каждую вершину представляют в роли промежуточной и через нее осуществляют модификацию пометки, т. е. на каждом шаге одну из вершин добавляют к множеству промежуточных вершин.
5. Аналогично рассмотренному ранее алгоритму Дейкстры, легко модифицировать таким образом, чтобы он возвращал не только длину кратчайшего пути, но и сам путь.
6. Для этого, сохранив преемственность, введем дополнительную матрицу G, в которой для каждой пары вершин u и v будем хранить номер первой промежуточной вершины кратчайшего пути из u в v.

У алгоритма Флойда есть замечательная особенность — проверять собственную применимость: алгоритм позволяет определить наличие циклов отрицательной длины в графе.

Основные понятия

1. **Постановка задачи**. Имеется ориентированный или неориентированный взвешенный граф, веса которого задаются **неотрицательными** вещественными числами. Требуется найти длины всех кратчайших расстояний между всеми парами вершин графа.
2. **Идея алгоритма**. Проводится расширение множества промежуточных вершин для путей, по которым ищутся кратчайшие расстояния.
3. Алгоритм начинает работу с матрицы весов (матрицы пометок $d(i, j)$), рассматриваемых как длины кратчайших путей от вершины v_i к вершине v_j, не имеющих ни одной промежуточной вершины, т. е. вес ребра (дуги) (v_i, v_j).
4. Далее поочередно каждую вершину представляют в роли промежуточной и через нее осуществляют модификацию пометки, т. е. на каждом шаге одну из вершин добавляют к множеству промежуточных вершин.
5. Аналогично рассмотренному ранее алгоритму Дейкстры, легко модифицировать таким образом, чтобы он возвращал не только длину кратчайшего пути, но и сам путь.
6. Для этого, сохранив преемственность, введем дополнительную матрицу G, в которой для каждой пары вершин u и v будем хранить номер первой промежуточной вершины кратчайшего пути из u в v.

У алгоритма Флойда есть замечательная особенность — проверять собственную применимость: алгоритм позволяет определить наличие циклов отрицательной длины в графе.

При наличии цикла отрицательной длины в матрице пометок $d(i, j)$ появятся отрицательные числа на главной диагонали.

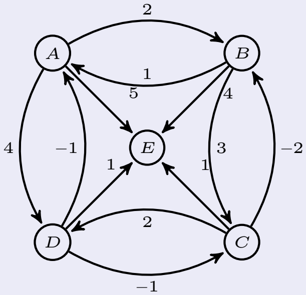
Алгоритм

Алгоритм 2.0. Флойда

```
// Инициализация
for i ← 1, n do
    for j ← 1, n do
        d(i, j) ← w(i, j)                    // Матрицу пометок заполняем весами ребер
        G(i, j) ← i                          // v_i — первая промежуточная вершина пути v_i → v_j
    end for
end for
// Основной алгоритм
// Внешний цикл идет по промежуточным вершинам k, внутри него для каждой
// пары вершин проверяется возможность улучшения их пометки через k
for k ← 1, n do
    for i ← 1, n do
        for j ← 1, n do
            if d(i, k) + d(k, j) < d(i, j) then
                d(i, j) ← d(i, k) + d(k, j)
                G(i, j) ← G(k, j)
            end if
        end for
    end for
end for
// Нахождение кратчайших путей для каждой пары вершин v_i, v_j
// L(i, j) —- список вершин кратчайшего пути v_i → v_j
for i ← 1, n do
    for j ← 1, n do
        L(i, j) ← ∅
        if d(i, k) ≠ ∞ then
            k = i                            // v_i — первая вершина пути L(i, j)
            do
                L(i, j) ← L(i, j) ∪ {k}
                k ← G(k, j)
            while k ≠ G(k, j)
            // Если промежуточная вершина подпути v_k → v_j совпала с его началом v_k,
            // то других промежуточных вершин на этом подпути нет
            L(i, j) ← L(i, j) ∪ {j}
            // Добавляем в список L(i, j) последнюю вершину пути v_i → v_j
        end if
    end for
end for
```

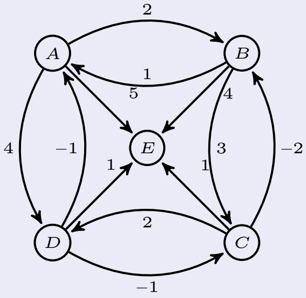
Пример. Исходные данные

Пример. Исходные данные



(a) *Исходный граф*

Пример. Исходные данные

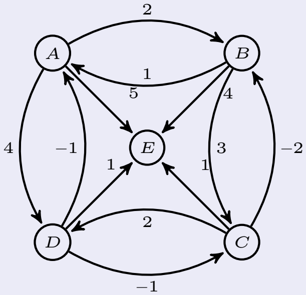


(a) Исходный граф

	A	B	C	D	E
A	0	2	∞	4	5
B	1	0	3	∞	4
C	∞	-2	0	2	1
D	-1	∞	-1	0	1
E	∞	∞	∞	∞	0

(b) Таблица весов графа

Пример. Исходные данные



(a) *Исходный граф*

	A	B	C	D	E
A	0	2	∞	4	5
B	1	0	3	∞	4
C	∞	-2	0	2	1
D	-1	∞	-1	0	1
E	∞	∞	∞	∞	0

(b) *Таблица весов графа*

	A	B	C	D	E
A	0	$2/A$	∞	$4/A$	$5/A$
B	$1/B$	0	$3/B$	∞	$4/B$
C	∞	$-2/C$	0	$2/C$	$1/C$
D	$-1/D$	∞	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(c) *Исходная матрица пометок*

Пример. Шаг 1

Пример. Шаг 1

- На первом шаге рассмотрим A в качестве промежуточной вершины.

Пример. Шаг 1

A На первом шаге рассмотрим A в качестве промежуточной вершины.

Обратим внимание на начальные вершины ребер, входящих в A (столбец A), в дальнейшем назовем их входящими вершинами, и на конечные вершины ребер, выходящих из A (строка A).

Пример. Шаг 1

A На первом шаге рассмотрим A в качестве промежуточной вершины.

Обратим внимание на начальные вершины ребер, входящих в A (столбец A), в дальнейшем назовем их входящими вершинами, и на конечные вершины ребер, выходящих из A (строка A).

Через ведущую вершину A можно улучшить пометки:

$$d(B,D) > d(B,A) + d(A,D) = \quad 1 + 4 \Rightarrow d(B,D) = 5$$
$$d(D,B) > d(D,A) + d(A,B) = -1 + 2 \Rightarrow d(D,B) = 1$$

Пример. Шаг 1

A На первом шаге рассмотрим A в качестве промежуточной вершины.

Обратим внимание на начальные вершины ребер, входящих в A (столбец A), в дальнейшем назовем их входящими вершинами, и на конечные вершины ребер, выходящих из A (строка A).

Через ведущую вершину A можно улучшить пометки:

$$d(B, D) > d(B, A) + d(A, D) = \quad 1 + 4 \Rightarrow d(B, D) = 5$$

$$d(D, B) > d(D, A) + d(A, B) = -1 + 2 \Rightarrow d(D, B) = 1$$

Протокол первого шага представлен в следующей таблице.

	A	B	C	D	E
A	0	$2/A$	∞	$4/A$	$5/A$
B	$1/B$	0	$3/B$	∞	$4/B$
C	∞	$-2/C$	0	$2/C$	$1/C$
D	$-1/D$	∞	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(a) *Исходная матрица пометок*

	A	B	C	D	E
A	0	$2/A$	∞	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	∞	$-2/C$	0	$2/C$	$1/C$
D	$-1/D$	$1/A$	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(b) *Протокол первого шага*

Пример. Шаг 2

Пример. Шаг 2

- Далее в качестве промежуточной вершины рассматриваем вершину B.

Пример. Шаг 2

B Далее в качестве промежуточной вершины рассматриваем вершину B.

Через вершину B улучшаем пометки:

$$d(A,C) > d(A,B) + d(B,C) = \quad 2 + 3 \Rightarrow d(A,C) = 5$$
$$d(C,A) > d(C,B) + d(B,A) = -2 + 1 \Rightarrow d(C,A) = -1$$

Пример. Шаг 2

Далее в качестве промежуточной вершины рассматриваем вершину B.

Через вершину B улучшаем пометки:

$$d(A,C) > d(A,B) + d(B,C) = \quad 2+3 \Rightarrow d(A,C) = 5$$

$$d(C,A) > d(C,B) + d(B,A) = -2+1 \Rightarrow d(C,A) = -1$$

Протокол второго шага представлен в следующей таблице.

	A	B	C	D	E
A	0	$2/A$	∞	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	∞	$-2/C$	0	$2/C$	$1/C$
D	$-1/D$	$1/A$	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(a) *Предыдущий шаг*

	A	B	C	D	E
A	0	$2/A$	$5/B$	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-1/D$	$1/A$	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(b) *Протокол второго шага*

Пример. Шаг 3

Пример. Шаг 3

- Следующей промежуточной вершиной будет вершина C.

Пример. Шаг 3

C Следующей промежуточной вершиной будет вершина C.

На этом шаге модифицируются пометки:

$$d(D,\,A) > d(D,\,C) + d(C,\,A) = -1 - 1 \Rightarrow d(D,\,A) = -2$$
$$d(D,\,B) > d(D,\,C) + d(C,\,B) = -1 - 2 \Rightarrow d(D,\,B) = -3$$
$$d(D,\,E) > d(D,\,C) + d(C,\,E) = -1 + 1 \Rightarrow d(D,\,E) = 0$$

Пример. Шаг 3

C Следующей промежуточной вершиной будет вершина C.

На этом шаге модифицируются пометки:

$$d(D, A) > d(D, C) + d(C, A) = -1 - 1 \Rightarrow d(D, A) = -2$$

$$d(D, B) > d(D, C) + d(C, B) = -1 - 2 \Rightarrow d(D, B) = -3$$

$$d(D, E) > d(D, C) + d(C, E) = -1 + 1 \Rightarrow d(D, E) = 0$$

Протокол третьего шага представлен в следующей таблице.

	A	B	C	D	E
A	0	$2/A$	$5/B$	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-1/D$	$1/A$	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(a) *Предыдущий шаг*

	A	B	C	D	E
A	0	$2/A$	$5/B$	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-2/C$	$-3/C$	$-1/D$	0	$0/C$
E	∞	∞	∞	∞	0

(b) *Протокол третьего шага*

Пример. Шаг 4. Кратчайшие пути

Пример. Шаг 4. Кратчайшие пути

D Следующий шаг — модификация через вершину D.

Пример. Шаг 4. Кратчайшие пути

D Следующий шаг — модификация через вершину D.

На этом шаге модифицируются три пометки:

$$d(A,B) > d(A,D) + d(D,B) = 4 - 3 \Rightarrow d(A,B) = 1$$
$$d(A,E) > d(A,D) + d(D,E) = 4 + 0 \Rightarrow d(A,E) = 4$$
$$d(A,C) > d(A,D) + d(D,C) = 4 - 1 \Rightarrow d(A,C) = 3$$

Пример. Шаг 4. Кратчайшие пути

D Следующий шаг — модификация через вершину D.

На этом шаге модифицируются три пометки:

$$d(A, B) > d(A, D) + d(D, B) = 4 - 3 \Rightarrow d(A, B) = 1$$

$$d(A, E) > d(A, D) + d(D, E) = 4 + 0 \Rightarrow d(A, E) = 4$$

$$d(A, C) > d(A, D) + d(D, C) = 4 - 1 \Rightarrow d(A, C) = 3$$

Протокол четвертого шага представлен в следующей таблице.

	A	B	C	D	E
A	0	$2/A$	$5/B$	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-2/C$	$-3/C$	$-1/D$	0	$0/C$
E	∞	∞	∞	∞	0

(a) Предыдущий шаг

	A	B	C	D	E
A	0	$1/D$	$3/D$	$4/A$	$4/D$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-2/C$	$-3/C$	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(b) Протокол четвертого шага

E На последнем шаге с промежуточной вершиной E модификаций пометок нет.

Пример. Шаг 4. Кратчайшие пути

D Следующий шаг — модификация через вершину D.

На этом шаге модифицируются три пометки:

$$d(A, B) > d(A, D) + d(D, B) = 4 - 3 \Rightarrow d(A, B) = 1$$

$$d(A, E) > d(A, D) + d(D, E) = 4 + 0 \Rightarrow d(A, E) = 4$$

$$d(A, C) > d(A, D) + d(D, C) = 4 - 1 \Rightarrow d(A, C) = 3$$

Протокол четвертого шага представлен в следующей таблице.

	A	B	C	D	E
A	0	$2/A$	$5/B$	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-2/C$	$-3/C$	$-1/D$	0	$0/C$
E	∞	∞	∞	∞	0

(a) Предыдущий шаг

	A	B	C	D	E
A	0	$1/D$	$3/D$	$4/A$	$4/D$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-2/C$	$-3/C$	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(b) Протокол четвертого шага

E На последнем шаге с промежуточной вершиной E модификаций пометок нет.

1 Найдем кратчайший путь от A до B. Имеем:

$$G(A, B) = D,\ G(D, B) = C,\ G(C, B) = C.$$

Пример. Шаг 4. Кратчайшие пути

D Следующий шаг — модификация через вершину D.

На этом шаге модифицируются три пометки:

$$d(A,B) > d(A,D) + d(D,B) = 4 - 3 \Rightarrow d(A,B) = 1$$
$$d(A,E) > d(A,D) + d(D,E) = 4 + 0 \Rightarrow d(A,E) = 4$$
$$d(A,C) > d(A,D) + d(D,C) = 4 - 1 \Rightarrow d(A,C) = 3$$

Протокол четвертого шага представлен в следующей таблице.

	A	B	C	D	E
A	0	$2/A$	$5/B$	$4/A$	$5/A$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-2/C$	$-3/C$	$-1/D$	0	$0/C$
E	∞	∞	∞	∞	0

(a) *Предыдущий шаг*

	A	B	C	D	E
A	0	$1/D$	$3/D$	$4/A$	$4/D$
B	$1/B$	0	$3/B$	$5/A$	$4/B$
C	$-1/B$	$-2/C$	0	$2/C$	$1/C$
D	$-2/C$	$-3/C$	$-1/D$	0	$1/D$
E	∞	∞	∞	∞	0

(b) *Протокол четвертого шага*

E На последнем шаге с промежуточной вершиной E модификаций пометок нет.

1 Найдем кратчайший путь от A до B. Имеем:

$$G(A,B) = D,\ G(D,B) = C,\ G(C,B) = C.$$

2 Таким образом кратчайший путь $A \to B$:

$$A \to D \to C \to B$$