Санкт-Петербургский Политехнический Университет

Высшая школа прикладной математики и вычислительной физики, ФизМех

01.03.02 Прикладная математика и информатика

Лабораторная работа №2

Дисциплина “Дискретная математика”

Тема “ Графы ”

Вариант “Алгоритм Флойда-Уоршалла”

Выполнил студент гр. 5030102/20201 Мелко Тимофей Андреевич

Поставленная задача

Реализовать алгоритм Флойда-Уоршалла для поиска кратчайших путей между всеми парами вершин взвешенного орграфа.

Используемый язык программирования

Python 3.12.6

Описание алгоритма Флойда-Уоршалла

Используемые обозначения в данном алгоритме:

1. W – исходный граф
2. p – количеств о вершин в графе
3. T – матрица кратчайших расстояний
4. P – Матрица путей

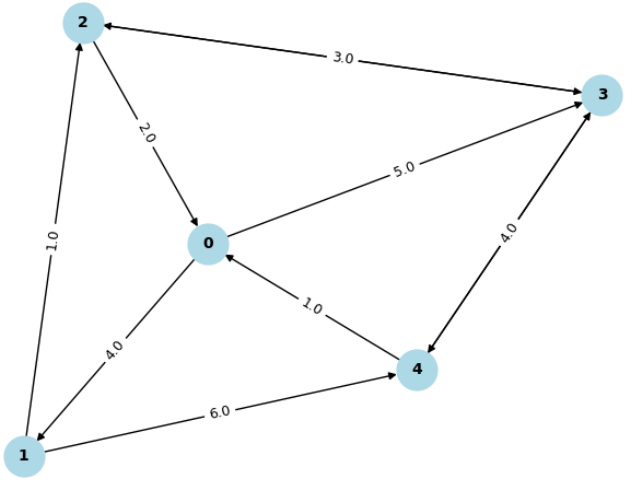
```
Функция Floyd_Warshall(W, p, T, P)
  // Инициализация матриц T и P
  Для i от 0 до p-1
    Для j от 0 до p-1
      T[i][j] ← W[i][j]  // Кратчайшие расстояния инициализируются значениями из W
      Если W[i][j] равно бесконечность
        P[i][j] ← inf  // Отсутствует путь
      Иначе
        P[i][j] ← i  // Устанавливаем предшественника для пути i -> j

  // Основной цикл алгоритма Флойда-Уоршалла
  Для k от 0 до p-1
    Для i от 0 до p-1
      Для j от 0 до p-1
        Если T[i][k] не равно бесконечность И T[k][j] не равно бесконечность
          Если T[i][j] > T[i][k] + T[k][j] // Проверка на более короткий путь
            T[i][j] ← T[i][k] + T[k][j]  // Обновление кратчайшего расстояния
            P[i][j] ← P[k][j]  // Устанавливаем предшественника

  // Проверка на наличие отрицательных циклов
  Для j от 0 до p-1
    Если T[j][j] < 0
```

Вернуть -1 // Найден отрицательный цикл

Пример работы алгоритма

Для примера рассмотрим орграф W, который зададим с помощью матрицы смежности, где inf отсутствие пути.

	0	1	2	3	4
0	0	4	inf	5	Inf
1	inf	0	1	inf	6
2	2	inf	0	3	Inf
3	inf	inf	1	0	2
4	1	inf	inf	4	0



1. Начальная инициализация

Запишем изначальную матрицу путей и длин путей

T =

	0	1	2	3	4
0	0	4	inf	5	Inf
1	inf	0	1	inf	6
2	2	inf	0	3	Inf
3	inf	inf	1	0	2
4	1	inf	inf	4	0

P =

	0	1	2	3	4
0	0	0	inf	0	Inf

1	inf	1	1	inf	1
2	2	inf	2	2	Inf
3	inf	inf	3	3	3
4	4	inf	inf	4	4

2. Рассмотри вершину 0 как промежуточную и получим

T =

	0	1	2	3	4
0	0	4	inf	5	Inf
1	inf	0	1	inf	6
2	2	6	0	3	Inf
3	inf	inf	1	0	2
4	1	5	inf	4	0

P =

	0	1	2	3	4
0	0	0	inf	0	Inf
1	inf	1	1	inf	1
2	2	0	2	2	Inf
3	inf	inf	3	3	3
4	4	inf	inf	4	4

3. Рассмотрим вершину 1 как промежуточную

T =

	0	1	2	3	4
0	0	4	5	5	10
1	inf	0	1	inf	6
2	2	6	0	3	12
3	inf	inf	1	0	2
4	1	5	6	4	0

P =

	0	1	2	3	4
0	0	0	1	0	1
1	inf	1	1	inf	1
2	2	0	2	2	1
3	inf	inf	3	3	3
4	4	0	1	4	4

4. Рассмотрим вершину 2 как промежуточную

T =

	0	1	2	3	4
0	0	4	5	5	7
1	inf	0	1	4	6
2	2	6	0	3	5
3	inf	inf	1	0	2
4	1	5	6	4	0

P =

	0	1	2	3	4
0	0	0	1	0	3
1	inf	1	1	2	1
2	2	0	2	2	3
3	inf	inf	3	3	3
4	4	0	1	4	4

5. Рассмотрим вершину 3 как промежуточную

T =

	0	1	2	3	4
0	0	4	5	5	7
1	inf	0	1	4	6
2	2	6	0	3	5
3	inf	inf	1	0	2

4	1	5	6	4	0

P =

	0	1	2	3	4
0	0	0	1	0	3
1	inf	1	1	2	1
2	2	0	2	2	3
3	inf	inf	3	3	3
4	4	0	1	4	4

6. Рассмотрим вершину 4 как промежуточную

T =

	0	1	2	3	4
0	0	4	5	5	7
1	7	0	1	4	6
2	2	6	0	3	5
3	3	7	1	0	2
4	1	5	6	4	0

P =

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

После прохождения алгоритмом получим матрицу кратчайший путей между всеми вершинами

T =

	0	1	2	3	4
0	0	4	5	5	7
1	3	0	1	4	6

2	2	6	0	3	5
3	3	7	1	0	2
4	1	5	5	4	0

Сложность алгоритма

Сложность алгоритма Флойда-Уоршелла заключается в необходимости рассчитать кратчайшие пути для всех пар вершин графа. В алгоритме 3 вложенных цикла перебирающие вершины графа. Таким образом количество операций для выполнения алгоритма O(p^3), где p – количество вершин в графе.

Входные и выходные данные

Входные данны.

Квадратная матрица n×n, где n — количество вершин в графе.

Элемент W[i][j]:

- Равен весу ребра из вершины i в вершину j, если такое ребро существует.
- Равен ∞ (или очень большому числу), если ребра между вершинами i и j нет.
- Равен 0, если i=j (вес петли на вершине равен нулю).

Выходные данные.

Матрица кратчайших расстояний T:

- Квадратная матрица n×n, где T[i][j] - вес кратчайшего пути между вершинами i и j.
- Если путь недостижим, T[i][j]=∞.

Матрица предшественников P:

- Квадратная матрица n×n, где P[i][j] хранит индекс вершины, непосредственно предшествующей вершине j на кратчайшем пути из i в j.
- Если путь недостижим, P[i][j]=∞.
- Если i=j, P[i][j]=i.

Область применимости

Алгоритм Флойда-Уоршелла лучше всего подходит для графов с малым и средним количеством вершин. Из-за сложности O(p^3), где p — число вершин, алгоритм не подходит для графов с большим количеством вершин, поскольку вычислительные затраты и объем памяти быстро увеличиваются с ростом графа. Входной граф не должен содержать отрицательных циклов, так как в этом случае задача поиска кратчайшего пути не имеет смысла. Так же алгоритм может обрабатывать как положительные, так и отрицательные веса.

Представление графов в программе

Для представления графа в программе я буду использовать матрицу смежности. Матрица смежности представляет собой двумерный массив, где каждый элемент T[i][j] хранит вес ребра от вершины i к вершине j. Доступ к данным в матрице занимает O(1) что делает возможным прямую

и быструю работу с каждой парой вершин. Поскольку алгоритм Флойда-Уоршелла требует многократного обновления расстояний между всеми парами вершин, матрица смежности хорошо подходит для этого. Так же матрица смежности позволяет удобно выполнять итерации без дополнительного поиска рёбер между вершинами, так как информация о связности вершин хранится напрямую в структуре данных.

Вывод

Алгоритм Флойда-Уоршалла позволяет найти кратчайшее расстояние между любыми двумя вершинами в графе, при этом веса ребер могут быть как положительными, так и отрицательными. Для графов большой размерности алгоритм может выполняться медленно из-за сложности O(p^3), где p – кол-во вершин.