

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”

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

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
```

```
         $T[i][j] \leftarrow W[i][j]$  // Кратчайшие расстояния инициализируются значениями из W
```

```
        Если  $W[i][j]$  равно бесконечность
```

```
             $P[i][j] \leftarrow 0$  // Отсутствует путь
```

```
        Иначе
```

```
             $P[i][j] \leftarrow i$  // Устанавливаем предшественника для пути  $i \rightarrow j$ 
```

```
// 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
```

```
                     $T[i][j] \leftarrow T[i][k] + T[k][j]$  // Обновление кратчайшего расстояния
```

```
                     $P[i][j] \leftarrow P[k][j]$  // Устанавливаем предшественника
```

```
// Проверка на наличие отрицательных циклов
```

```
Для j от 0 до p-1
```

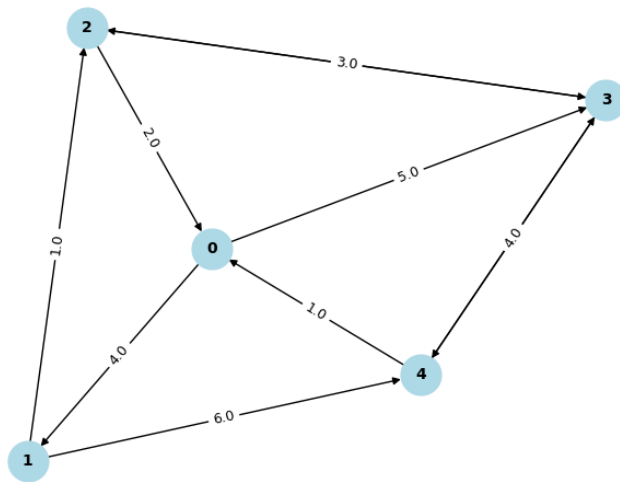
```
    Если  $T[j][j] < 0$ 
```

```
        Вернуть -1 // Найден отрицательный цикл
```

Пример работы алгоритма

Для примера рассмотрим оргграф W

	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



После прохождения алгоритмом получим

	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

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.