

URL на GitHub

<https://github.com/rrenatovich/FloudWarshallPython.git>

Основная идея

Разбиение задачи поиска минимальных путей на фазы/шаги

перед k фазой считаем, что в матрице расстояний сохранены длины кратчайших путей (где внутренние вершины от 1 до $k - 1$)

Т.е. перед шагом k величина $D[i][j]$ это кратчайший путь из вершины i в вершину j при условии, что можно заходить в вершины с номерами до $k - 1$

Пусть мы находимся на k фазе и хотим пересчитать матрицу D для $k + 1$. Могут возникнуть случаи:

1. Кратчайший путь из вершины i в вершину j , которому разрешено дополнительно проходить через вершины $\{1, 2, \dots, k\}$, совпадает с кратчайшим путём, которому разрешено проходить через вершины множества $\{1, 2, \dots, k - 1\}$

Тогда матрица D не изменится

2. Новый путь оказался лучше

Разобьём путь вершиной k :

1. Одна $i \rightarrow k$

2. Вторая $k \rightarrow j$

Длина каждой из этих половинок была посчитана на шаге $k - 1$ и мы можем взять сумму $d[i][k] + d[k][j]$ это и будет длина нового кратчайшего пути

Получается что на k фазе можем найти кратчайший путь между всеми парами вершин:

```
graph[i][j] = min(graph[i][j], graph[i][k] + graph[k][j])
```

```
Example 1
Shortest Paths Graph for A
[0, 5, 15, 10, 10, 10, 0]
[20, 0, 10, 5, 0, 15, 2]
[30, 35, 0, 15, 123, 0, 1]
[15, 20, 5, 0, 123, 2, 2]
Execution time: 0.003003358840942383
```