

# ОТЧЕТ О ПРОДЕЛАННОЙ РАБОТЕ №1

Вариант № 7. Алгоритм Флойда- Уоршелла  
Студент Газзаев Алан- Барсаг Валерьевич 411117, группа М4130

## Ссылка на исходный код:

[https://github.com/alanrbtx/evolve\\_lab1/blob/main/lab1.ipynb](https://github.com/alanrbtx/evolve_lab1/blob/main/lab1.ipynb)

## Описание работы

Алгоритм Флойда (алгоритм Флойда–Уоршелла) – это алгоритм нахождения длин кратчайших путей между всеми парами вершин во взвешенном ориентированном графе. Работает корректно, если в графе нет циклов отрицательной величины, а в случае, когда такой цикл есть, позволяет найти хотя бы один такой цикл

## Сложность алгоритма

Внешний цикл в этом алгоритме последовательно перебирает все вершины, затем пытается улучшить пути из  $V$ , разрешив им проходить через выбранную вершину. Упростим этот алгоритм, избавившись от «трехмерности» массива  $A$ : будем только хранить значение кратчайшего пути из  $V$  в  $A[i][j]$ , а при улучшении пути будем записать новую длину пути также в  $A[i][j]$ . Также изменим определение цикла по переменной  $k$ , заменив значение  $k-1$  на  $k$ . Сложность такого алгоритма равна  $O(n^3)$

## Тесты и сравнение производительности алгоритмов

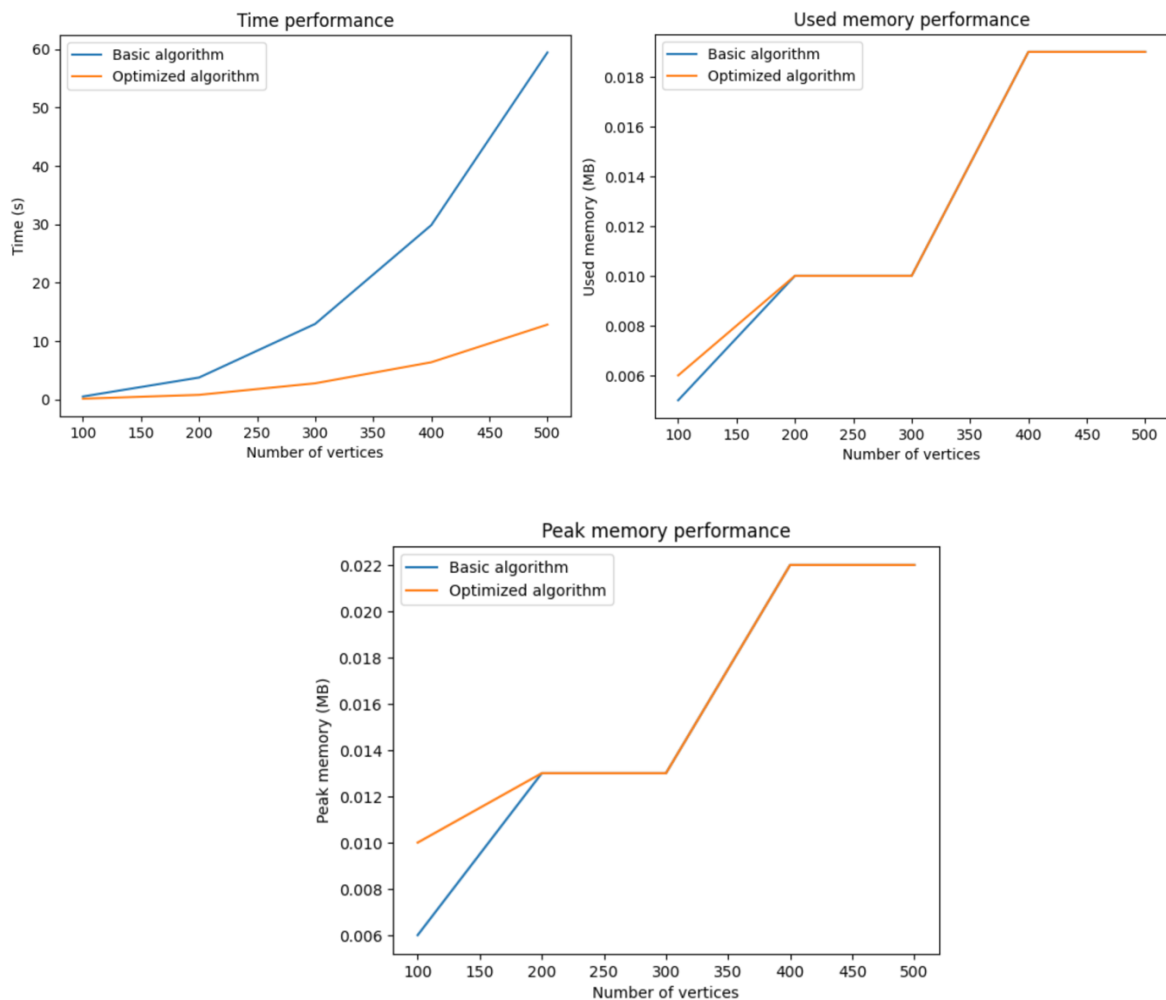
Базовый алгоритм

	Time	Used memory	Peak memory
Number of vertices			
100	0.481	0.005	0.006
200	3.733	0.010	0.013
300	12.924	0.010	0.013
400	29.852	0.019	0.022
500	59.435	0.019	0.022

## Оптимизированный алгоритм

	Time	Used memory	Peak memory
Number of vertices			
100	0.108	0.006	0.010
200	0.765	0.010	0.013
300	2.743	0.010	0.013
400	6.352	0.019	0.022
500	12.800	0.019	0.022

## Графики



Алгоритм оптимизировался следующим способом: попробуем ускорить алгоритм Флойда-Уоршелла, добавив условие, чтобы не пересчитывать пути, если дистанция между вершинами не изменилась после предыдущей итерации. Это позволит сократить количество лишних операций. Алгоритм

действительно стал работать быстрее, при этом утилизация памяти осталась такой же.

## **Выводы**

Было реализовано два варианта алгоритма Флойда- Уоршелла, также были проведены тестовые замеры производительности обоих алгоритмов и построены графики, на которых наглядно видны различия в производительности.