# Тепляков Владислав БПИ2310
# SET 6
# Задача A2

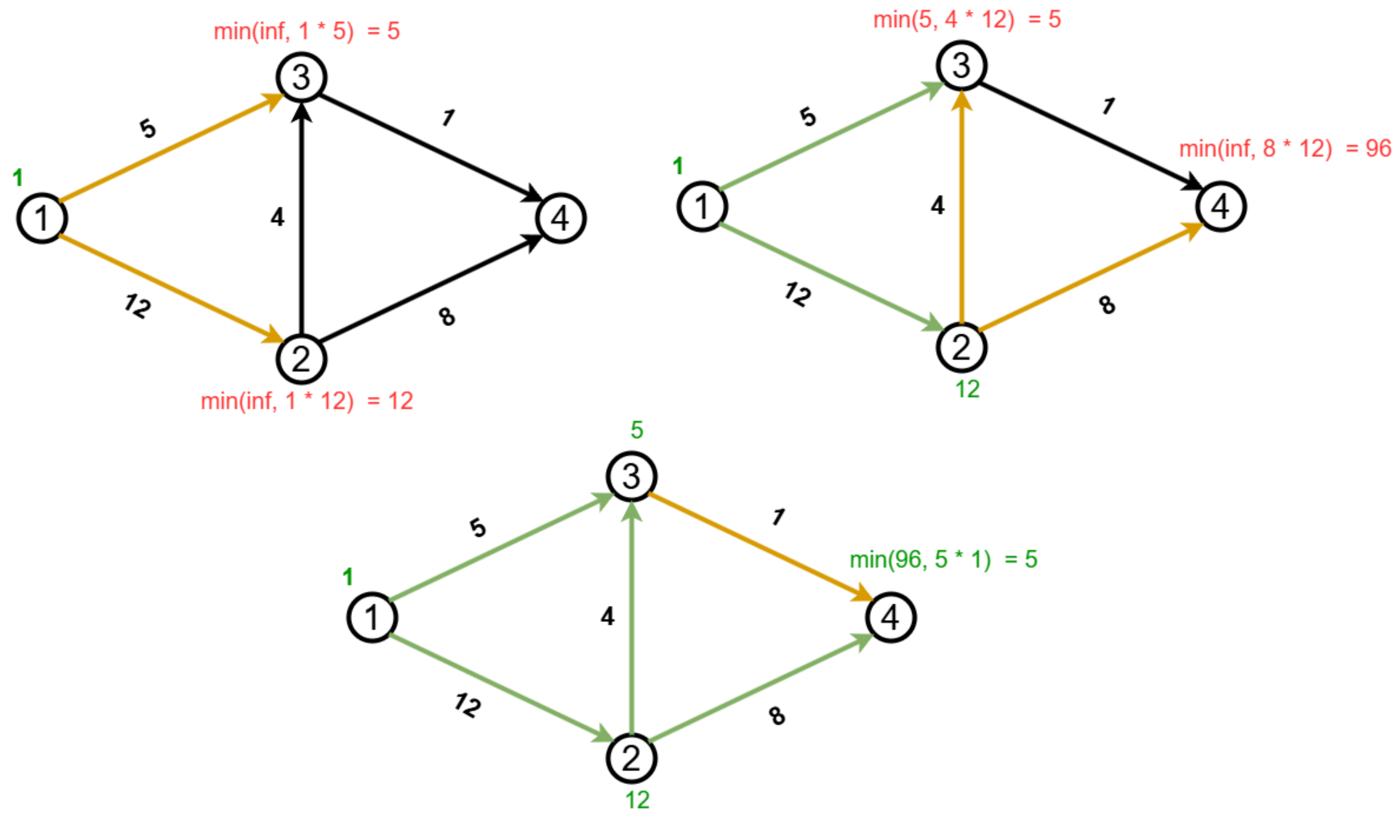
## Часть 1

В алгоритме Дейкстры мы начинаем с первой вершины и присваиваем ей значение 1. Все остальные вершины мы помечаем как бесконечно большие.

На каждом шаге алгоритма мы выбираем вершину с наименьшим значением. Затем мы обновляем расстояние от текущей вершины до соседних, беря минимум из значений в соседних вершинах и умножая его на вес пути от текущей вершины до соседней.

Этот алгоритм будет работать правильно для графов с положительными весами рёбер, потому что произведение положительных чисел сохраняет порядок. Также важно, чтобы в графе не было цикла с рёбрами с весом от 0 до 1, иначе мы будем бесконечно улучшать значение.

Вот как работает алгоритм: Изначально все вершины, кроме “1”, имеют значение бесконечности.



## Часть 2

Для каждой пары вершин i и j, где i ≠ j и dist[i][j] < ∞, мы проверяем, есть ли ребро i → j (или j → i для неориентированного графа) на кратчайшем пути. Если расстояние от i до j равно сумме расстояний от i до других вершин и веса ребра между ними, то это ребро может быть частью графа. Для каждой пары i и j мы перебираем все возможные вершины k и проверяем это условие. Если оно выполняется, то мы добавляем ребро k → j в граф и присваиваем ему вес, равный dist[k][j]. Важно отметить, что граф может быть как ориентированным, так и неориентированным. В случае ориентированного графа мы добавляем направленные рёбра, а в неориентированном — рёбра в обе стороны: k → j и j → k.

Если матрица кратчайших путей (dist) корректно отображает кратчайшие пути в графе без циклов с отрицательной длиной, то восстановление графа возможно. Однако если в матрице есть несоответствия или она была создана для графа с циклами или ошибками, то восстановить граф не получится.

## Часть 3

Проблема заключается в том, что цикл по переменной k должен быть внешним, а не внутренним. Это связано с тем, что на каждом шаге k мы обновляем расстояния через промежуточную вершину k. Если порядок циклов в алгоритме Флойда-Уоршелла нарушен, то промежуточные вершины не учитываются должным образом. В архиве есть пример трассировки, который демонстрирует, как алгоритм работает некорректно.

## Часть 4

Существует такой граф, в котором ребра могут быть частью кратчайших путей в обе стороны, если веса заданы таким образом, чтобы минимизировать эти пути. Алгоритмы поиска кратчайших путей работают хорошо, если веса положительны или если в графе нет отрицательных циклов. Однако, если такие циклы присутствуют, алгоритмы могут давать неверные результаты.

Важно подчеркнуть, что алгоритмы, которые требуют отсутствия отрицательных циклов, такие как алгоритм Дейкстры и алгоритм Флойда-Уоршелла, будут работать корректно только в том случае, если веса неотрицательны.

Пример графа: Вершины a, b, c, d. Ребра: a → c (вес 1), c → d (вес 2), d → b (вес 1), b → d (вес 1), d → c (вес 2), c → a (вес 1). Кратчайший путь a → b: a → c → d → b (вес 1 + 2 + 1 = 4). Кратчайший путь b → a: b → d → c → a (вес 1 + 2 + 1 = 4). Ребро c → d лежит на пути a → b, а d → на пути b → a