

# Описание программы

## Описание работы алгоритма

Рассмотрим как работает алгоритм построения транзитивного замыкания с помощью алгоритма Уоршелла:

Алгоритм работает с матрицей смежности. Идея алгоритма Уоршелла состоит в расширении множества промежуточных вершин по следующему правилу: на каждом шаге в рассмотрение добавляется одна новая вершина, после чего достижимости вершин пересчитываются "через нее".

То есть если  $w$  — промежуточная вершина, то достижимость вершины  $v$  из вершины  $u$  через  $w$  пересчитывается по правилу:

$$d(u, v) := \max(d(u, v), d(u, w) \cdot d(w, v))$$

т. е. не меняется, если  $v$  достижима из  $u$ , и меняется (с 0 на 1), если достижимости до введения промежуточной вершины  $w$  не было, а  $w$  достижима из  $u$  и  $v$  достижима из  $w$ . Таким образом, после  $k$  шагов будут соединены те вершины, которые достижимы по путям, проходящим только через первые  $k$  вершин (кроме первой и последней).

Также можно записать эту формулу в логических операциях (т.к. ребра принимают значения 0 или 1):  $d(u, v) := d(u, v) \vee (d(u, w) \wedge d(w, v))$

Пусть бинарное отношение  $f$  задано на множестве  $M$ , тогда запишем псевдокод для алгоритма Уоршелла:

```
// Инициализация матрицы (если ее нет изначально)
for each u in M:
    for each v in M:
        d(u,v) = f(a,b) // Инициализация значения "ячейки" матрицы
                           смежности

//Основной алгоритм
for each w in M:
    for each u in M:
        for each v in M:
            d(u,v) = d(u,v) or (d(u, w) and d(w, v))
```

Таким образом можем заметить что алгоритм выполняется за  $O(N^3)$ , где N - кол-во вершин в матрице.

## Визуализация

Планируется, что на экране после запуска программы пользователю предоставится возможность задать матрицу смежности и на её основе будет сгенерирован граф, как метод визуализации матрицы. В зависимости от изменения матрицы (добавления/удаления рёбер или вершин) будет происходить соответствующие изменения в графе.

В ходе работы алгоритма планируется возможность выбора режима с помощью "переключателей":

- Обычный режим - при запуске алгоритма моментально появляется итоговая матрица смежности с транзитивным отношением и соответствующий ей изменённый граф.
- Режим показа работы - при запуске алгоритма граф и матрица будут преобразовываться постепенно с подробной визуализацией изменения. Также данный режим делится на "ручной" и "автоматический". Эти подрежимы будут описаны ниже.

## Визуализация матрицы

Матрица смежности до начала работы алгоритма будет выглядеть примерно следующим образом:

	A	B	C	D
A	-	0	0	0
B	0	-	0	0
C	0	0	-	0
D	0	0	0	-

На некотором шаге алгоритма матрица будет отображаться примерно следующим образом:

	A	B	C	D
A	-	0	0	0
B	0	-	0	0
C	0	0	-	0
D	0	0	0	-

(где условно красным обведена главная для этого шага вершина (ту, через которую смотрим можно ли пройти), а жёлтым обрабатываемая на данном шаге ячейка)

После добавления очередного ребра матрица смежности будет выглядеть примерно следующим образом:

	A	B	C	D
A	-	1	0	0
B	0	-	0	0
C	1	1	-	0
D	0	0	0	-

(на каждом шаге вновь добавленное ребро будет отображаться в графе жёлтым цветом, а рёбра породившие его, через которые мы смогли пройти, - синим)

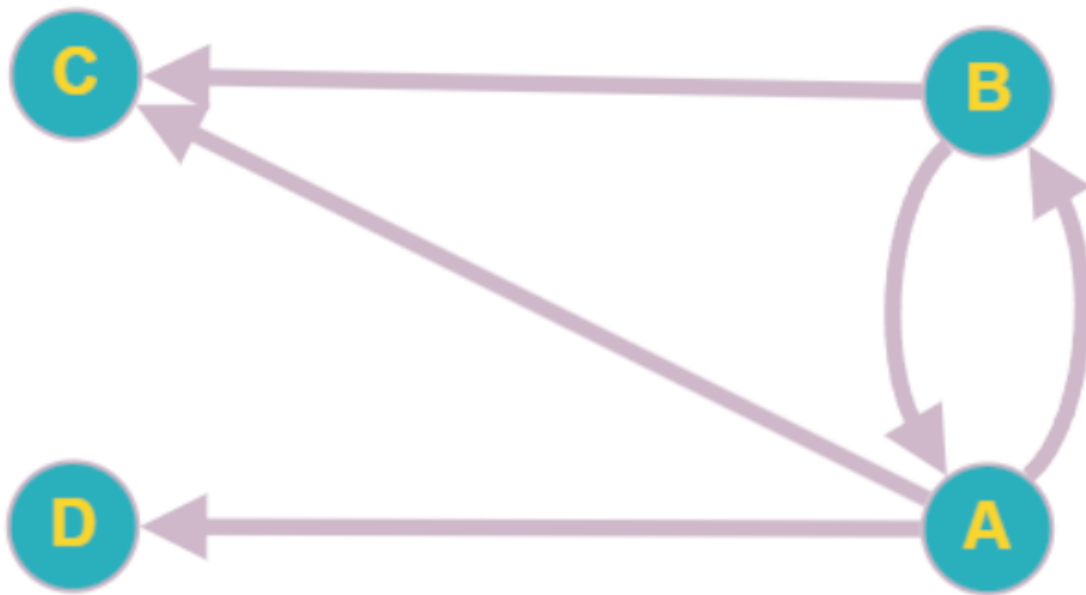
На каждом из последующих шагов рёбра, которые были добавлены ранее в алгоритме, будут отмечаться в матрице смежности условно зелёным цветом. После работы

алгоритма все добавленные рёбра в матрице будут выделены этим цветом:

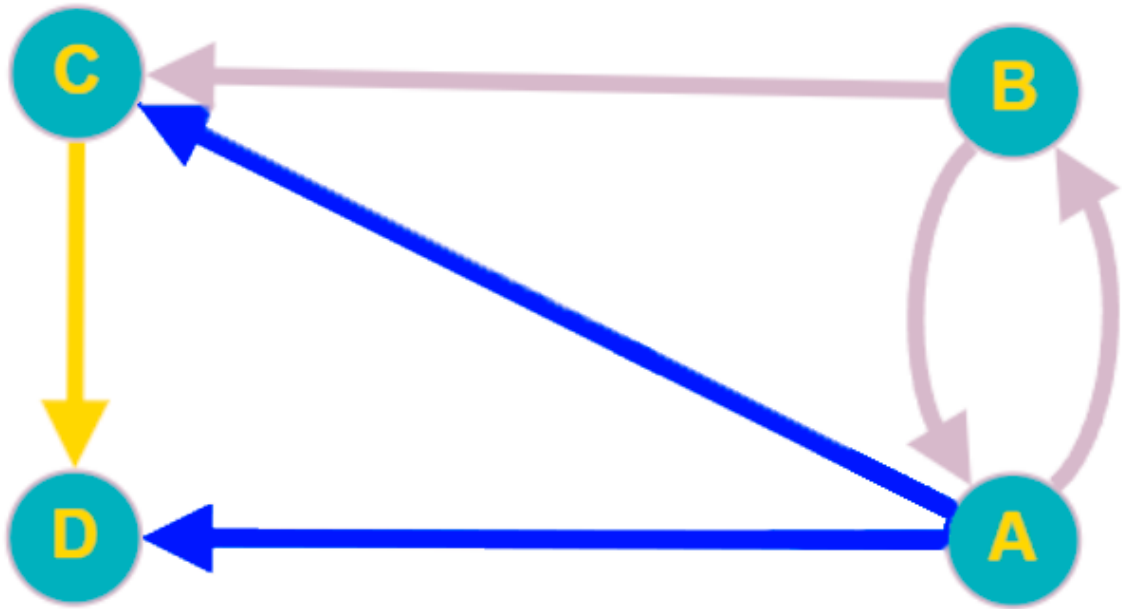
	A	B	C	D
A	-	1	0	0
B	0	-	0	1
C	1	1	-	0
D	1	1	0	-

## Визуализация графа

Относительно матрицы смежности некоторого графа будет строиться его визуальное отображение, которое будет выглядеть примерно так:

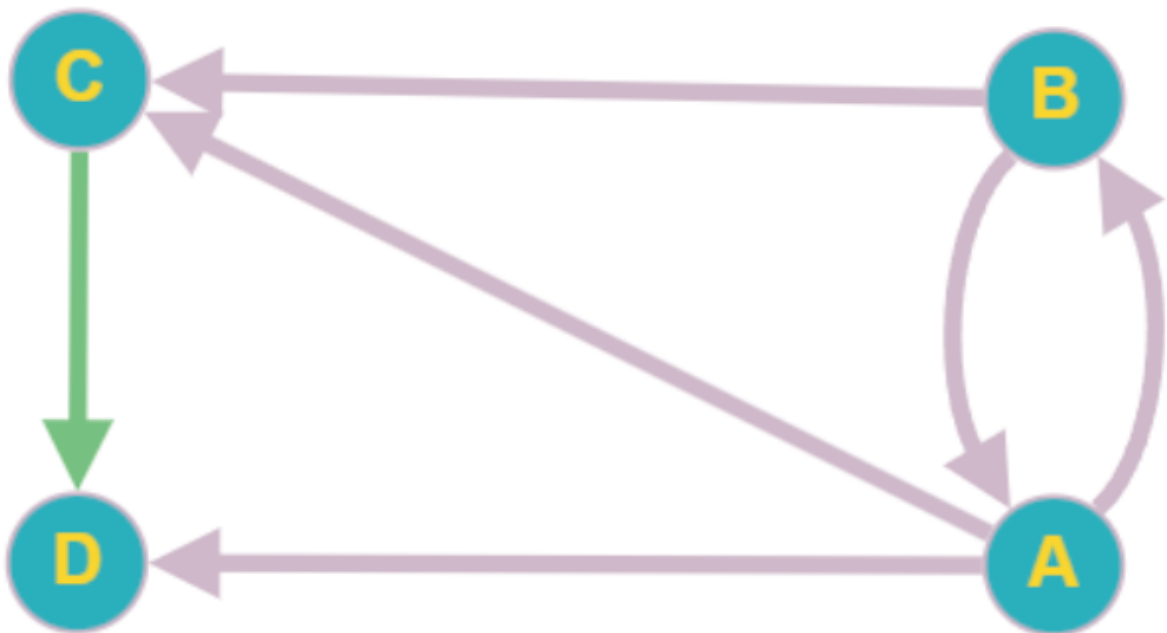


Во время работы алгоритма на каждой итерации в случае, если было добавлено новое ребро, оно будет выделено определённым цветом, а также будет выведено сообщение вида "добавлено ребро  $V \rightarrow U$ ". Также будут выделены рёбра, из-за которых данное ребро появилось. Примерное отображение графа при добавлении нового ребра:



(на каждом из шагов рёбра, которые были добавлены ранее в алгоритме, будут окрашиваться условно в жёлтый цвет, а рёбра, которые породили это ребро, будут окрашиваться условно в синий)

После работы алгоритма все добавленные рёбра будут выделены отдельным цветом, что поможет оценить работу алгоритма и посмотреть, как изменился граф:



## Дополнительные взаимодействия пользователя с интерфейсом

Помимо изложенной информации о визуализации планируется, что пользователю на главном экране будут доступны кнопки вида:

- Добавления/удаления вершин в матрице - пользователю будут предоставлены две кнопки: добавить, удалить, нажав на которые добавляется следующая вершина или удаляется последняя добавленная вершина;
- Кнопка загрузки матрицы смежности из файла - пользователю выведется отдельное меню для указания файла с матрицей смежности. При этом поведение данной кнопки подразумевает полное удаление старой матрицы смежности;
- Кнопка генерации случайного матрицы смежности - пользователю выведется отдельное меню для задания параметров генерации случайного матрицы смежности (точное количество вершин). При этом поведение данной кнопки подразумевает полное удаление старой матрицы смежности;
- Выбор режима: обычный/показ работы
- Для режима показа работы будет выбор режима показа пошаговой работы алгоритма (ручной/автоматический)
  - "Ручной" предполагает, что переход к следующему шагу будет осуществлен только при нажатии пользователем кнопки "малый шаг" или "средний шаг" или "большой шаг", которые делают шаг в одном из циклов алгоритма(в меньшем, среднем и большем соответственно);
  - "Автоматический" предполагает, что переход к следующему шагу будет осуществляться каждую секунду (можно будет указать время самому, например в диапазоне от 0.1 секунды до 10 секунд с помощью ползунка);
- Кнопка запуска алгоритма Уоршелла.

## План разработки и распределение ролей

Приблизительный план разработки будет разделен на следующие этапы:

- **Создание прототипа:** предполагает создание готового, но не рабочего (или частично рабочего) интерфейса приложения: визуализация матрицы с рабочими для ввода ячейками, строящийся для данной матрицы граф, наличие перечисленных кнопок, но без работоспособности. Предполагаемая дата завершения 02.07.2024.
- **Создание первой версии приложения:** предполагает доработку прототипа, в котором к UI элементам будет добавлена логика, добавятся возможности генерации матрицы с помощью файла или случайным образом и алгоритм можно запустить в

обычном режиме (без режима показа работы алгоритма). Предполагаемая дата завершения 04.07.2024.

- **Создание второй версии приложения:** предполагает доработку первой версии до состояния, в котором алгоритм работает в двух режимах (обычном режиме и режиме показа работы с фиксированным малым шагом в ручном режиме и с интервалом в 1 секунду в автоматическом режиме (без возможности выбора времени)). Предполагаемая дата завершения 07.07.2024.
- **Создание финальной версии приложения и написание отчета:** предполагает доработку режима показа работы алгоритма в ручном и автоматическом режиме, исправление возможных ошибок и написания отчета для разработанного приложения. Предполагаемая дата завершения 08.07.2024.

Далее будут описаны конкретные задачи, которые будут выполнены на каждом этапе разработки, и распределение этих задач по участникам команды.

#### **Задачи для написания прототипа приложения:**

- Разработка UI интерфейса приложения - создание разметки и добавление всех основных UI элементов программы. Также данная задача подразумевает разработку "системы оповещений" в интерфейсе для вывода ошибок и информации о работе алгоритма. (Данную задачу выполняет Пикалов Илья);
- Разработка отображения матрицы смежности графа - добавление отдельного UI элемента с возможностью задания матрицы. Подразумевается, что в этой матрице будет возможность задать значения каждой ячейки и возможность добавлять/удалять вершины. (Данную задачу выполняет Птицын Денис);
- Разработка визуализации матрицы с помощью графа - добавление отдельного UI элемента с возможностью отображения матрицы смежности в виде графа. Подразумевается, что граф будет отображаться автоматически при задании матрицы смежности, а также изменение графа при изменении матрицы смежности. (Данную задачу выполняет Затримайлов Даниил).

#### **Задачи для написания первой версии приложения:**

- Доработка UI интерфейса программы - добавление к UI элементам логики взаимодействия (Данную задачу выполняет Пикалов Илья);
- Импорт матрицы смежности из файла - добавление возможности загрузки матрицы смежности из файла. Подразумевается также обработка всевозможных исключений, которые могут появиться из-за чтения файла (Данную задачу выполняет Птицын Денис);
- Генерация случайной матрицы смежности - добавление возможности генерации случайно матрицы смежности с возможностью задать конкретное количество вершин

(Данную задачу выполняет Птицын Денис);

- Реализация алгоритма Уоршелла - подразумевается, что после запуска алгоритма можно увидеть изменения, произошедшие в матрице и графе (Данную задачу выполняет Затримайлов Даниил);

#### **Задачи для написания второй версии приложения:**

- Добавление автоматического режима показа работы алгоритма - подразумевается, что данный режим должен работать по той логике, которая описывалась ранее, но только с шагом в 1 секунду без возможности выбора времени (Данную задачу выполняет Затримайлов Даниил);
- Добавление ручного режима показа работы - подразумевается, что данный режим должен работать по той логике, которая описывалась ранее, но только с фиксированным маленьким шагом. Также подразумевает раскраску новых и участвующих в алгоритме рёбер в графе и ячеек матрицы (Данную задачу выполняет Пикалов Илья);
- Добавление вывода лога - в отдельном консольном приложении должен быть предусмотрен вывод лога о работе алгоритма (Данную задачу выполняет Птицын Денис).

#### **Задачи для написания финальной версии приложения:**

- Доработка автоматического/ручного режима до полной функциональности - для автоматического режима возможность выбора времени, а для ручного - возможность выбора шага (в соответствии с описанием этих режимов выше) (Данную задачу выполняет Затримайлов Даниил);
- Доработка вывода лога в соответствии с доработкой режимов первой задачи финальной версии (Данную задачу выполняет Птицын Денис);
- Написание отчёта о проделанной работе (Данную задачу выполняет Пикалов Илья);