

## Требования (спецификация):

Состав спецификации:

1. Постановка задачи.
2. Формат входных данных.
3. Формат выходных данных.
4. Дополнительные возможности графического интерфейса.

1. Программа иллюстрирует алгоритм Борушки на заданном графе;

2. Пользователю предлагается два варианта задания графа:

а) С помощью матрицы смежности, записанной в файл: в интерфейсе программы пользователю будет предложена опция загрузки этого файла, программа извлечет из него данные, проверит их корректность, и построит соответствующий матрице граф. Вершины графа после загрузки будут распределяться по кругу;

б) С помощью инструментов графического интерфейса:

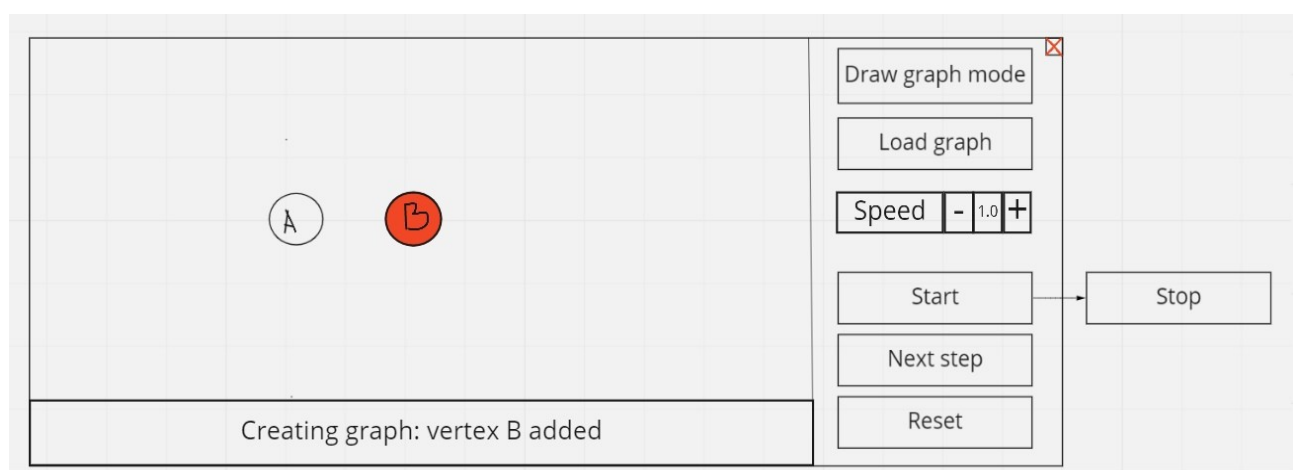


Рисунок 1 — Эскиз графического интерфейса

\*Создание графа происходит в режиме «Draw graph mode» (он активируется после нажатия соответствующей кнопки в окне программы): добавление и удаление вершин происходит при нажатии левой и правой клавиши мыши соответственно.

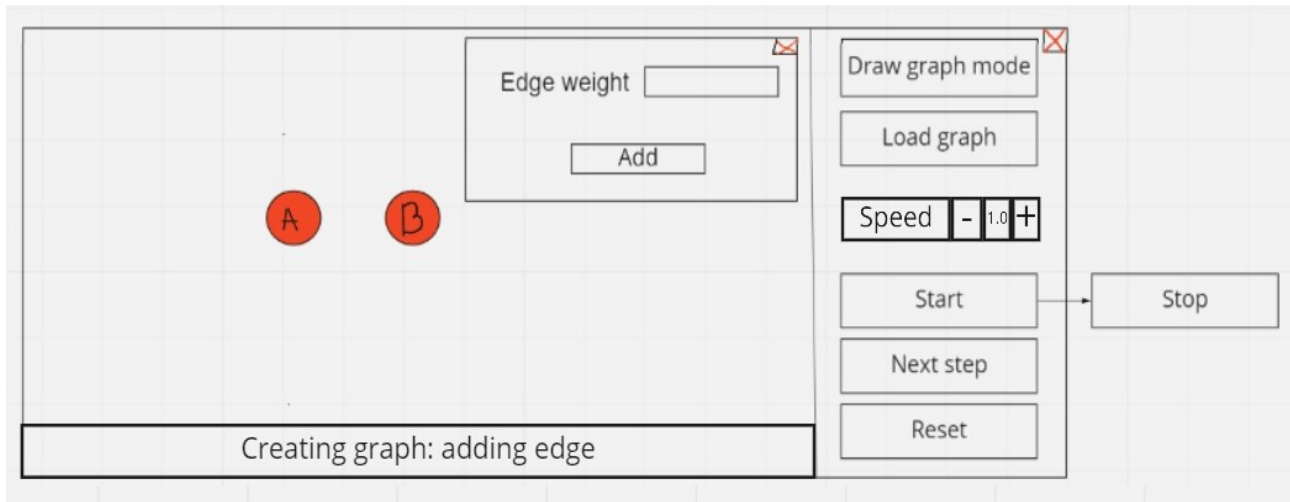


Рисунок 2 — Эскиз добавление и соединение вершин

\*Для того чтобы соединить две вершины, нужно в «Draw graph mode» выбрать их по очереди с помощью левой клавиши мыши, после чего откроется диалоговое окно, в котором будет предложено указать вес нового ребра и подтвердить его добавление (как и вершины, ребра так же удаляются нажатием на них правой клавишей мыши).

### 3. Формат выходных данных:

После нажатия кнопки «Start» начинается пошаговая визуализация работы алгоритма. На каждом шаге различные компоненты связности выделяются каждая своим цветом. Выбранное на каждом «маленьком» шаге ребро окрашивается в красный цвет. Остановить визуализацию на конкретном шаге

можно будет с помощью кнопки «Stop». Также, для перемещения между шагами, будет создана кнопка «Next step»;

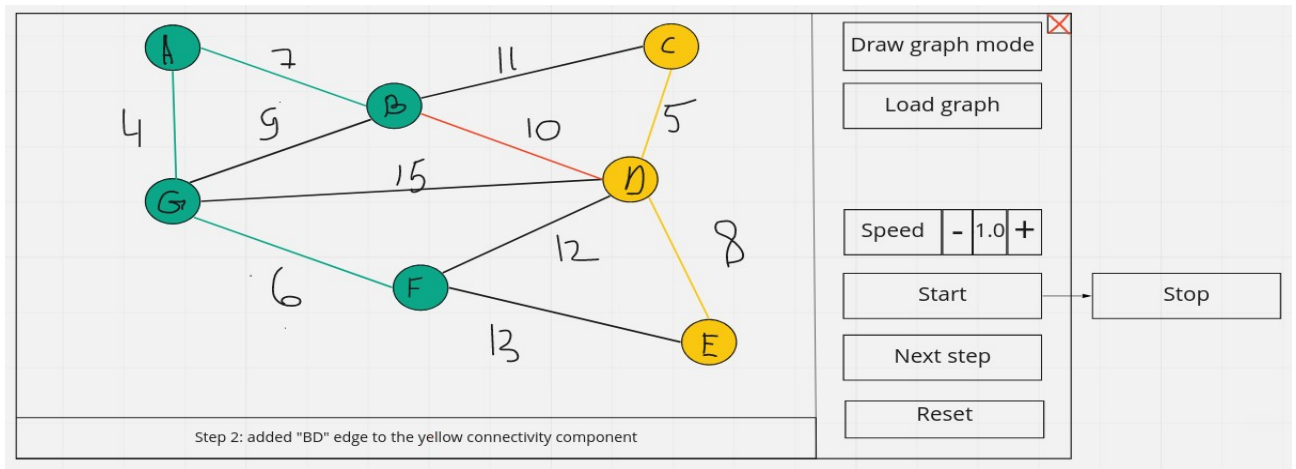


Рисунок 3 — Эскиз визуализации шага алгоритма Борувки

Кнопка «Next step» будет перемещать алгоритм на один «маленький шаг»: шаг, при котором соединяются две компоненты связности;

Также в приложении есть текстовое поле, в котором отображаются шаги алгоритма: «маленькие» и «большие» («маленький» шаг — шаг, при котором ребром соединяются две компоненты связности, «большой» шаг — шаг, при котором число компонент связности уменьшается как минимум вдвое);

К тому же, минимальное остовное дерево будет записано в виде матрицы в текстовом поле в конце работы алгоритма. Информацию из текстового поля можно будет копировать. По этому полю можно будет передвигаться сверху-вниз и снизу-вверх с помощью колеса мыши.

#### 4. Дополнительные возможности графического интерфейса.

\*Вне режима Draw graph mode можно будет передвигать вершины графа (зажав левую клавишу мыши), а также перемещаться по плоскости рисования;

\*Скорость выполнения шагов алгоритма возможно регулировать с помощью кнопок «+» и «-», одно нажатие на одну из этих кнопок соответственно

увеличивает или уменьшает скорость на 0.5 секунды (на панели, находящейся между ранее упомянутыми кнопками будет выводиться текущая скорость алгоритма). Стандартная скорость алгоритма — 1.0 секунды.

### **Уточнения после сдачи прототипа:**

К первой версии программы будет изменен формат представления результата работы алгоритма в текстовом поле: вместо матрицы смежности, будет выводиться список всех ребер, составляющих минимальное остовное дерево. Помимо этого, будет изменено представление матрицы смежности графа внутри файла: теперь, при вводе матрицы, пользователю обязательно будет требоваться указать названия вершин выше столбцов этой матрицы (опционально можно также указать названия вершин слева от строк), либо же пользователь сможет указать только нижнетреугольную матрицу.

A B C	A B C	A B C	A B C
A - 3 2	- 3 2	-	A -
B 3 - 1	3 - 1	3 -	B 3 -
C 2 1 -	2 1 -	2 1 -	C 2 1 -

Рисунок 4-Различные способы задания матрицы смежности в файле.

### **Уточнения после сдачи первой версии:**

Во второй версии программы будет дополнительно добавлена кнопка «Заново», которая будет переводить граф в начальное состояние для повторного запуска алгоритма. Также будет немного подкорректирован интерфейс загрузки графа из файла: при повторной загрузке по умолчанию будет выбираться папка, из которой был взят файл в прошлый раз. Также кнопка «Сбросить» будет переименована в кнопку «Очистить холст».



## **План разработки:**

1. План
2. Распределение бригады

## **План разработки:**

**1.1** «Прототип» должен включать в себя логику и минимальный графический интерфейс: холст, кнопки «Load graph» и «Start», первая — загружает граф из файла и отображает его на холсте, вторая — выводит минимальное остовное дерево графа в виде матрицы в текстовом поле, в текстовом поле на данном этапе будет реализован только вывод матрицы минимального остовного дерева.

\*Прототип будет готов к 06.07.2022.

**1.2** Первая версия должна включать весь функционал прототипа, однако кнопка «Start» получит новый функционал — помимо вывода матрицы в текстовое поле, она будет отображать на холсте минимальное остовное дерево графа. Будет добавлена кнопка «Reset», которая будет удалять граф с холста. Будет добавлена кнопка «Next step» - пошагового (по «маленьким» шагам) решения (в соответствии со спецификацией). На каждом шаге алгоритма вершины и рёбра будут раскрашиваться в соответствии со спецификацией.

\*Первая версия будет готова к 08.07.2022

**1.3** Вторая версия должна включать весь функционал первой версии, кнопки регулирующие скорость работы алгоритма, а также возможность задания графа с помощью мышки и холста (которая будет доступна через кнопку «Draw graph

mode»), возможность перемещения вершин и камеры относительно холста (всего графа целиком), а также текстовое поле получит новый функционал: в нем будет выводиться информация обо всех «маленьких» и «больших» шагах. Также кнопка «Start» при нажатии будет меняться на кнопку «Stop», останавливающую текущую работу алгоритма. Кнопка «Stop» при нажатии будет меняться обратно на кнопку «Start». Также будет добавлена регулировка скорости: кнопками “+” и “-”, а также с помощью текстового поля, в котором отображается текущая скорость. Кнопка «Reset» изменит название на «Clear canvas» и будет очищать холст (удалять граф). Будет добавлена кнопка «Again», которая будет перезапускать алгоритм (то есть отменять текущий прогресс алгоритма, как будто алгоритм и не запускался). При повторном выборе файла в окне выбора будет появляться папка, из которой был взят файл в прошлый раз.

\*Вторая версия будет готова к 11.07.2022

### **Распределение бригады:**

1) На логику будет выделено два человека: Корсунов Антон и Самулевич Василий: Антон будет разрабатывать структуру данных и интерфейс логики (для взаимодействия с графикой), Василий будет разрабатывать алгоритм.

2) На GUI будет выделен один человек — Сабанов Петр. Петр напишет графическое представление для прототипа, также в его обязанности в ходе дальнейшей разработки входит написание холста и возможностей взаимодействия с ним мышью.

\*В дальнейшем (после разработки прототипа) к Петру присоединятся другие члены команды, в обязанности которых будет входить разработка структуры данных для графического отображения графа, а также связь графического представления с логикой.

\*Распределение может корректироваться по ходу разработки.