

## Требования (спецификация):

Состав спецификации:

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

1. Программа иллюстрирует алгоритм Борувки 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:

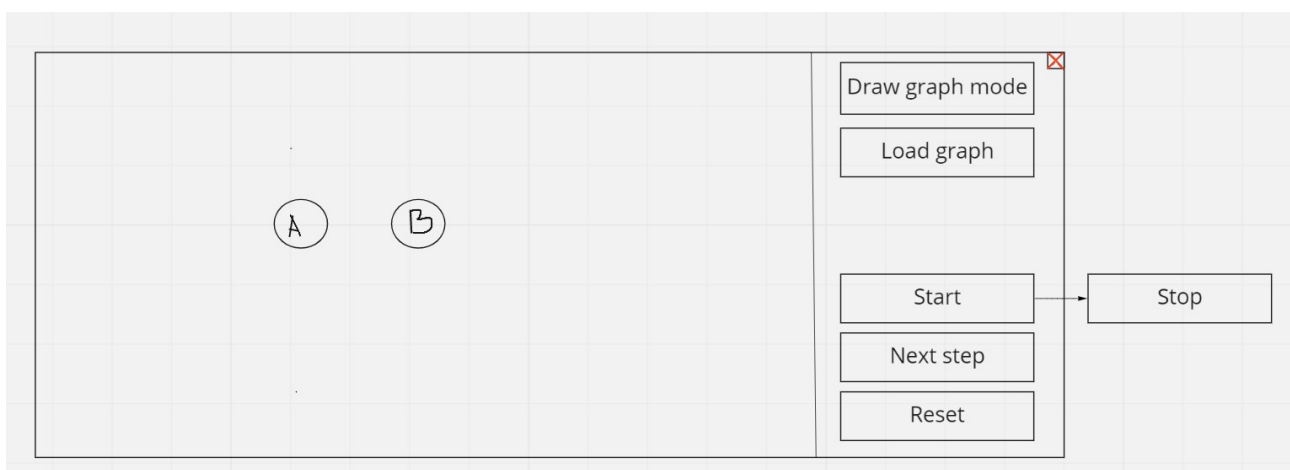


Рисунок 1 — Эскиз графического интерфейса

\*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.

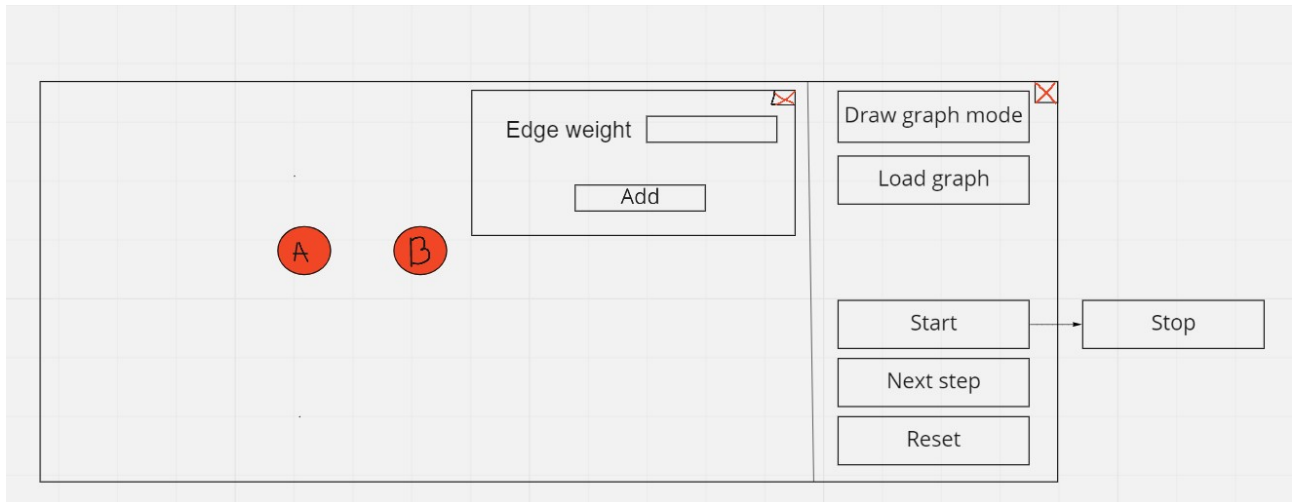


Рисунок 2 — Эскиз добавление и соединение вершин

\*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. Остановить визуализацию на конкретном шаге можно будет с помощью кнопки «Stop». Также, для перемещения между шагами, будет создана кнопка «Next step».

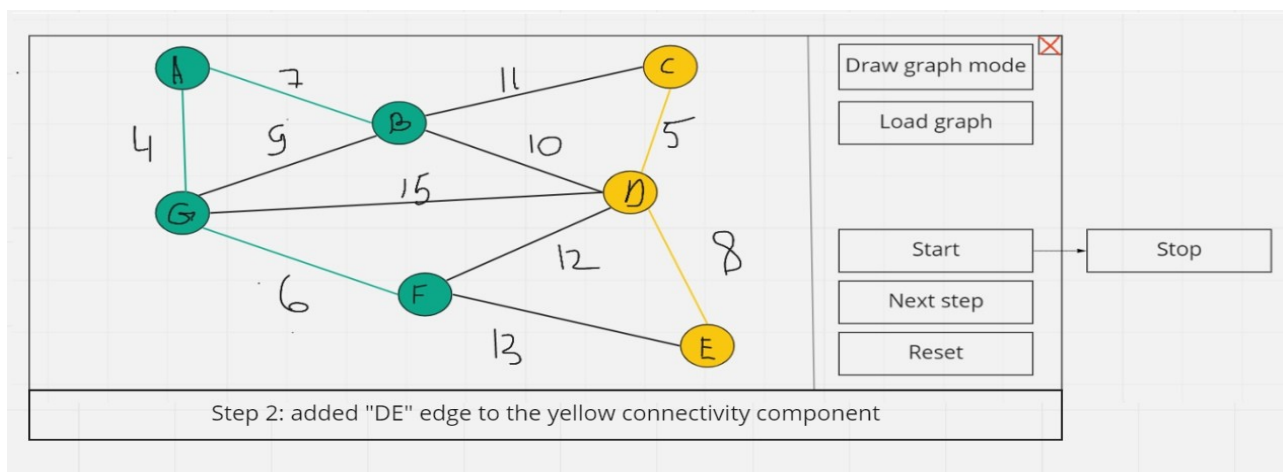


Рисунок 3 — Эскиз визуализации шага алгоритма Борувки

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).

#### 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.

## **План разработки:**

### 1. План

### 2. Распределение бригады

1. К версии «Прототип» (к 4-ому июля) планируется реализовать логику алгоритма и минимальную визуализацию: холст, кнопки: «загрузка из файла», «применить алгоритм», первая загружает граф из файла и отображает его на холсте, вторая отображает на холсте минимальное остовное дерево графа.

## **План разработки:**

1) «Прототип» должен включать в себя логику и минимальный графический интерфейс: холст, кнопки: «загрузка из файла», «применить алгоритм», первая загружает граф из файла и отображает его на холсте, вторая отображает на холсте минимальное остовное дерево графа.

2) Первая версия должна включать весь функционал прототипа, а также возможность пошагового (по «маленьким» шагам) решения (в соответствии со спецификацией), возможность перемещения вершин и камеры относительно холста (всего графа целиком), раскраску вершин и рёбер на шагах алгоритма, а также текстовое поле снизу от холста, в котором будет выводиться информация обо всех «маленьких» и «больших» шагах («маленький» шаг — шаг, при котором ребром соединяются две компоненты связности, «большой» шаг — шаг, при котором число компонент связности уменьшается как минимум вдвое).

3) Вторая версия должна включать весь функционал первой версии, а также возможность задания графа с помощью мышки и холста.

2. 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.

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 меняться по ходу разработки.