

## Спецификация

Задан исходный ориентированный граф  $G = (V, E)$ , где  $V$  – количество вершин в графе, для данного графа, в котором каждая вершина пронумерована от 0 до  $V$ . Необходимо реализовать и визуализировать алгоритм топологической сортировки для данного графа. После выполнения алгоритма будет получен отсортированный граф.

Программа будет предоставлять пользователю графический интерфейс. Входные данные будут считываться либо из файла формата .txt, либо с помощью специального виджета в окне программы. Ввод данных происходит следующим образом: вводится пара натуральных чисел через пробел, где первое число является начальной вершиной ребра, а второе – конечной. При чтении из файла, данные должны быть в том же формате. После запуска программы, пользователь наблюдает меню программы. Примерный эскиз окна представлен на рис. 1.

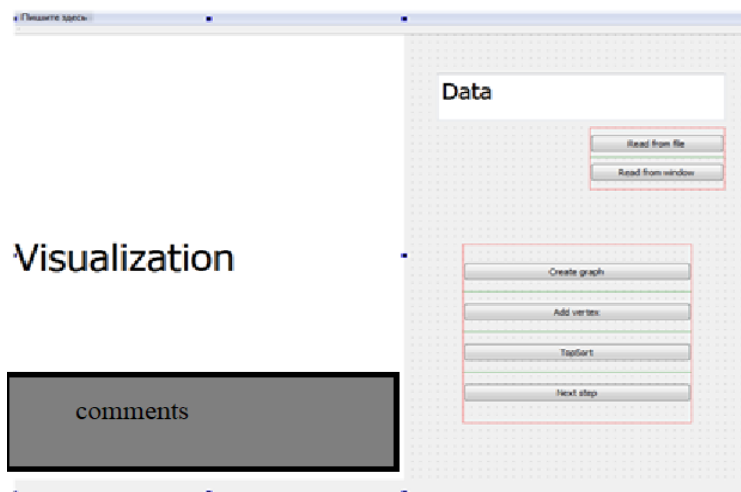


Рисунок 1 – Эскиз пользовательского интерфейса.

В редактор теста с пометкой «Data» (см. рис. 1) будут вводиться вершины. В специальном виджете с пометкой «Visualization» будет отображаться граф, при нажатии кнопки «Create graph», кликом будет добавляться вершина при нажатии на окно, где будет отображаться граф, отсортированный граф будет отображаться при нажатии «TopSort» и будет

показываться каждая итерация при нажатии «Next step». При отображении отсортированного графа вершины идут слева направо в соответствии с отношением частичного порядка. Ребра не должны сливаться.

С помощью данного приложения пользователь сможет считать с файла или ввести самостоятельно граф, который ему нужно будет отсортировать. Во время сортировки под окном визуализации в окне «comments» будет показываться какой шаг алгоритма был выполнен и почему. Например: если вершина серая — найден цикл, топологическая сортировка невозможна, будет сообщаться об ошибке и выводиться цикл, если вершина белая – красим ее в серый, если зеленая – просмотр вершины окончен. Древесные ребра будут краситься в синий цвет. Первая вершина цикла будет краситься в красный цвет вместе с ребром, по которому в нее был совершен второй переход. Проверка ацикличности и сама топологическая сортировка будут основаны на поиске в глубину

### **План разработки**

3.07 – согласование спецификации и плана разработки.

5.07 – сдача прототипа.

8.07 – реализация алгоритма, визуализация графа и отсортированного графа.

10.07 – визуализация алгоритма с возможностью ввода графа через графический интерфейс и перемещения вершин с покраской древесных рёбер и вершин, в которых зацикливается граф.

12.07 – разрешение проблемы сливающихся ребер при выводе отсортированного графа проекта сдача отчёта.

## Распределение обязанностей

ФИ	Область работы
Левкович Д.	Реализация алгоритма
Лосев М.	Реализация графического интерфейса и визуализации графа, тесты
Медведев И.	Реализация связи алгоритма с графическим интерфейсом, отчёт

## Уточнение спецификации

В ходе разработки прототипа был принят ряд изменений в интерфейсе программы. Кнопка «Add Edge» была перемещена под окно визуализации для более удобного использования. Кнопки «Read from file» и «Make graph» были помещены под окно ввода данных. Также кнопка «Make graph» стала выполнять функцию кнопки «Read from field». Кнопки «Next step» и «Run alg» были помещены вниз окна, правее окошка с комментариями. Изменения интерфейса представлены на рис. 2.

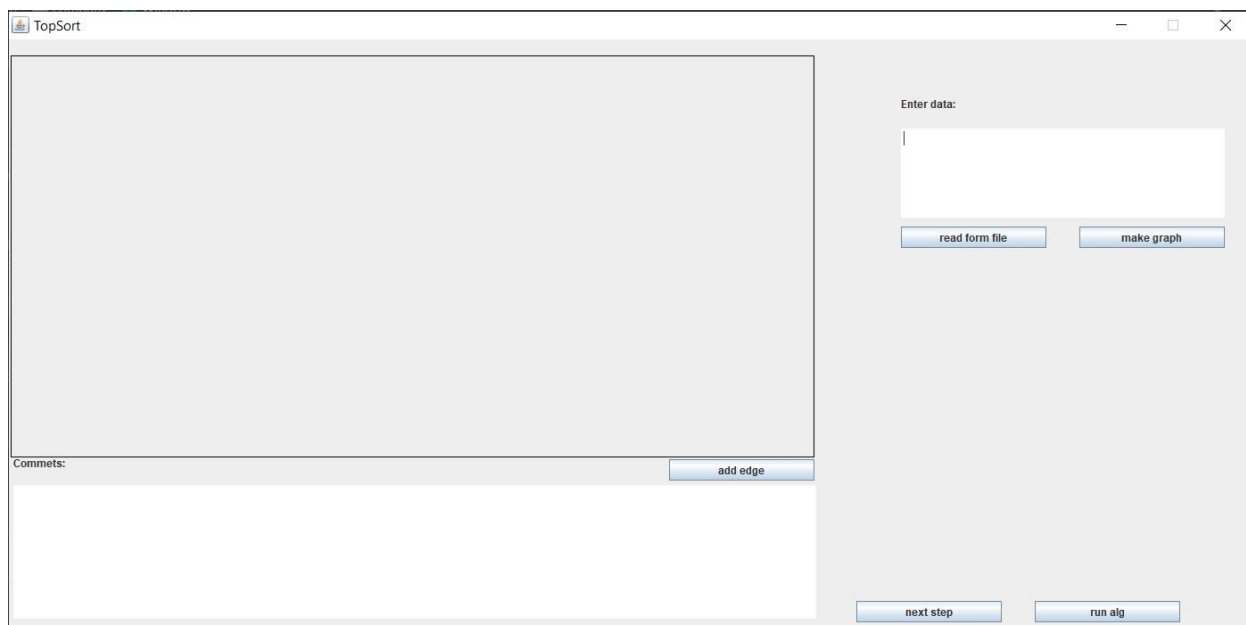


Рисунок 2 – Изменение интерфейса программы.

## Уточнение спецификации после первой итерации

В ходе разработки первой версии было принято добавить переносы строки при чтении из файла, для удобства чтения и проверки данных, добавить удаление ребра нажатием правой кнопкой мыши на соответствующее ребро, добавить кнопку для перехода в начало алгоритма и рисовать между соседними вершинами отсортированного графа прямые стрелочки.

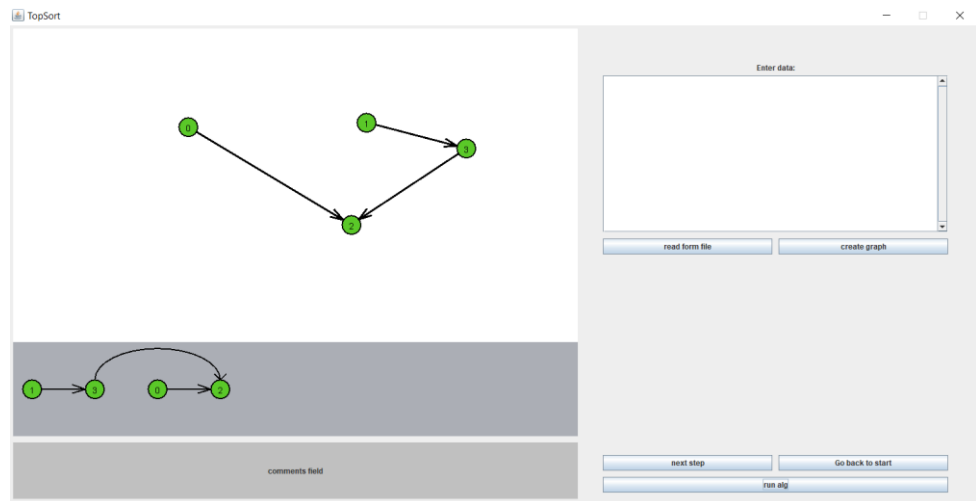


Рисунок 3 – Модификация интерфейса после первой итерации

## Уточнение спецификации после второй итерации

После сдачи второй итерации было решено внести следующие изменения:

- Добавить кнопку отмены последнего действия при вводе графа через графический интерфейс.
- Лог комментариев с прокруткой. 1)
- Окраску последней кликнутой вершины как выбранной другим цветом. 2)
- Последнее добавленное ребро в графе окрашивать другим цветом.
- Добавить прокрутку для отсортированного графа. 3)
- Объяснить порядок вершин в ответе. 4)
- Рядом с вершиной подписать номер выхода из нее. 5)

- Исправить перекраску древесных ребер: ребра, по которым еще не было возврата, окрашивать цветом, отличным от цвета ребер, по которым возврат уже был.
- Объяснить, как программа поняла, что обнаружен цикл, и выводить цикл в комментариях.