

СПЕЦИФИКАЦИЯ

Программа реализует пошаговую визуализацию алгоритма Ярника-Прима-Дейкстры поиска минимального остовного дерева в графе. На вход программе подается неориентированный граф, который задается графически, или в текстовом виде (список ребер и их весов). После завершения алгоритма полученное остовное дерево отображается на главном окне, также присутствует возможность записи полученного результата в текстовый файл. В ходе выполнения алгоритма пользователь может видеть предыдущее и текущее состояние программы, при этом добавленные в ответ, обрабатываемые на данном шаге и оставшиеся вершины и ребра отмечены разными цветами.

Главное окно программы (рисунок 1) представляет собой окно, содержащее 3 кнопки для ввода графа: ввод из файла (открывается диалоговое окно), ввод с клавиатуры (открывается окно с полем для ввода текста) и кнопку графического ввода (открывается окно графического ввода графа). При неправильном формате ввода программа показывает сообщение с ошибкой, пользователь может ввести граф еще раз. Также окно содержит кнопку сохранения графа в файл и кнопку запуска алгоритма, по нажатию на которую открывается окно, отображающее состояние программы, главное окно скрывается. Также окно содержит два поля, отображающих введенный граф и полученный в результате работы алгоритма граф.

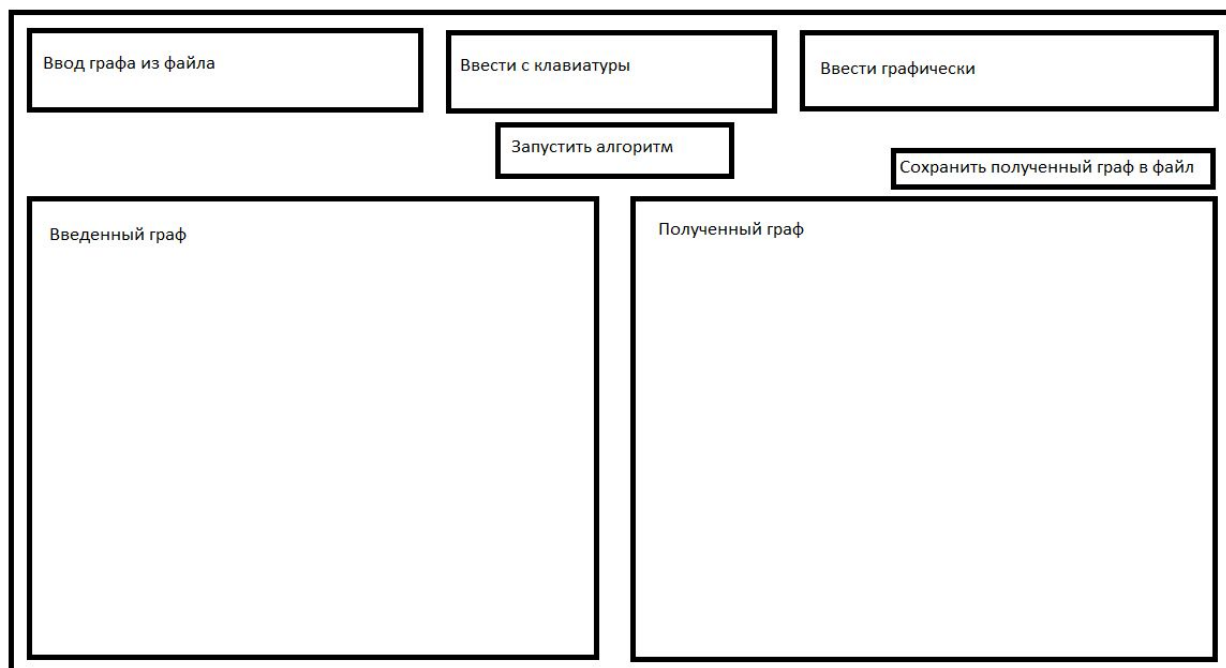


Рисунок 1 - Главное окно программы

Окно, отображающее состояние программы (рисунок 2), содержит два поля, в которых отображается состояние алгоритма до и после выполнения текущего шага. В нижней части поля вывода предыдущего состояния расположено поле для вывода комментариев по работе алгоритма. Ребра и вершины, добавленные в дерево, окрашены зеленым цветом, ребра и вершины, которые рассматриваются на данном шаге, окрашены синим цветом, остальные ребра и вершины окрашены черным цветом. В нижней части окна содержатся кнопки “Шаг вперед” и “Шаг назад”, по нажатию на которые программа переходит к следующему шагу алгоритма или возвращается на шаг назад. Программа не может вернуться назад более, чем на 32 шага. Также внизу содержится кнопка “Прервать алгоритм”, по нажатию на которую закрывается окно с состоянием программы и отображается главное окно, при этом поле с результатом работы остается пустым. Также внизу расположен таймер, по истечении времени которого автоматически выполняется следующий шаг алгоритма. По умолчанию таймер выставлен на 10 секунд, время можно изменить или остановить

таймер. При этом кнопка остановки таймера меняется на кнопку возобновления таймера.

The image shows a graphical user interface window with a black border. It is divided into several sections:

- Previous state:** A large rectangular area on the left side, labeled "Предыдущее состояние" (Previous state) in the top-left corner.
- New state:** A large rectangular area on the right side, labeled "Новое состояние" (New state) in the top-left corner.
- Comment field:** A horizontal rectangular area below the two state panels, labeled "Поле для комментариев" (Field for comments).
- Control bar:** A row of five buttons at the bottom of the window:
 - Прервать алгоритм** (Stop algorithm)
 - Шаг назад** (Step back)
 - Шаг вперед** (Step forward)
 - Пауза** (Pause)
 - Таймер** (Timer)

Рисунок 2 - Окно, отображающее состояние программы

Окно графического ввода (рисунок 3) содержит рабочую область, пару переключателей (радиокнопок) режима: “Редактирование вершин” и “Редактирование ребер”, а также кнопки применить и отменить. В рабочей области отображается граф. При активированном режиме редактирования вершин: клик в пустую область добавляет новую вершину, клик по вершине - удаляет вершину и инцидентные ребра. При активированном режиме редактирования ребер: поочередные клики на 2 вершины добавляет между ними ребро, если такого ребра еще нет, или удаляет ребро, если оно существует.

Область для ввода графа

Режим:

☒ Редактирование вершин

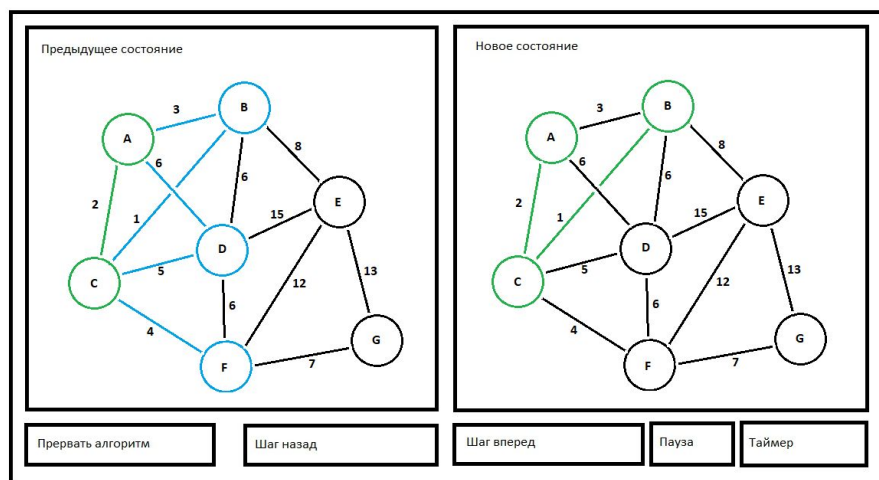
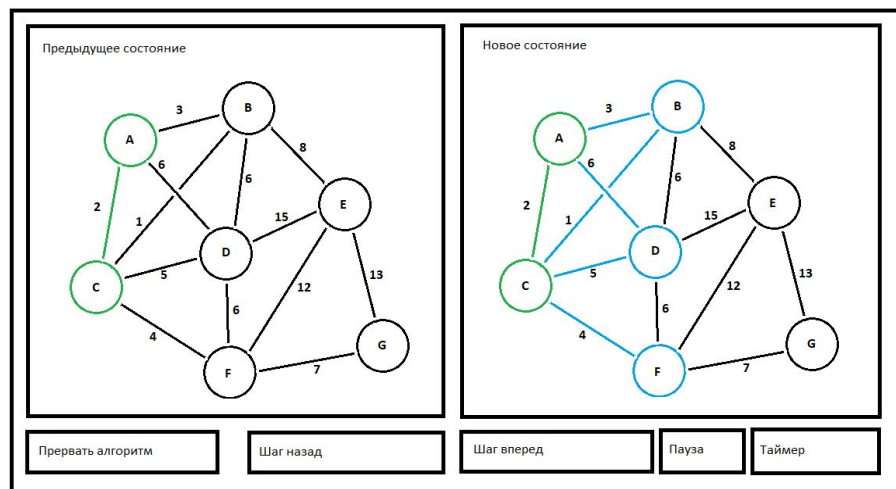
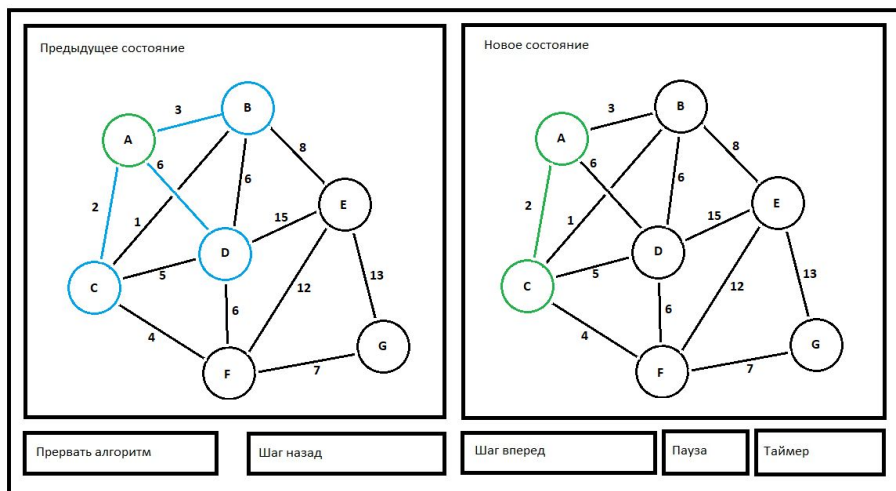
☐ Редактирование ребер

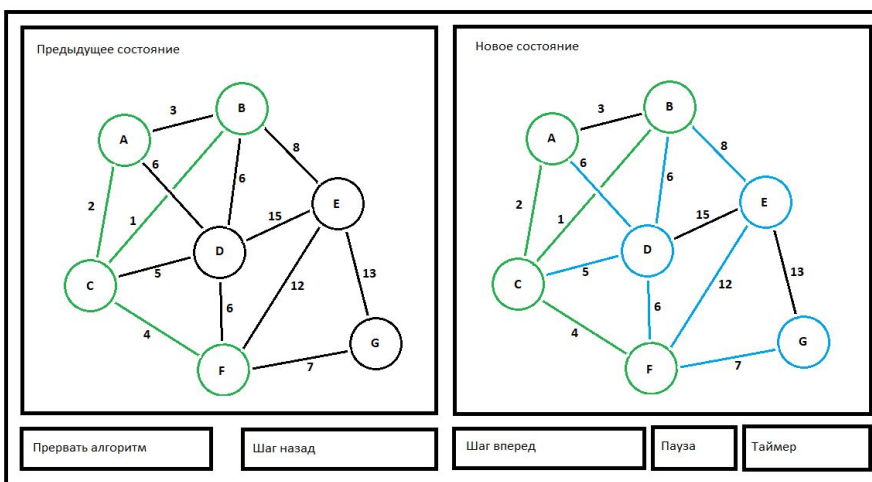
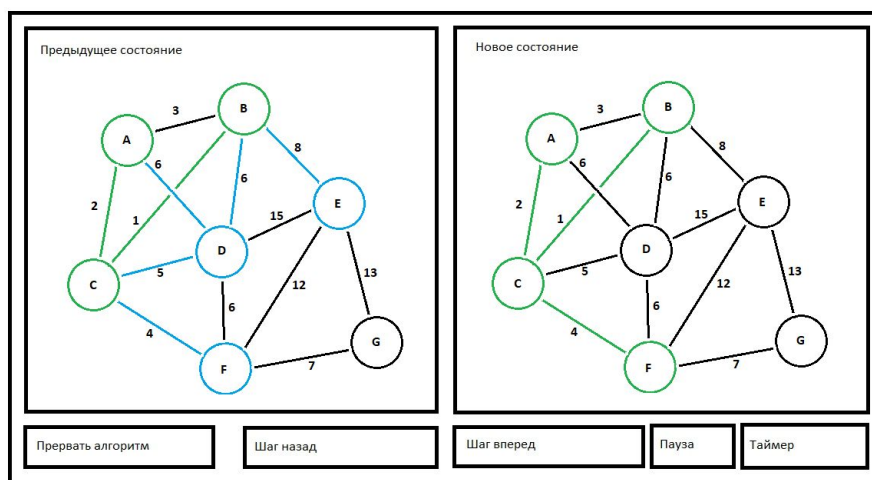
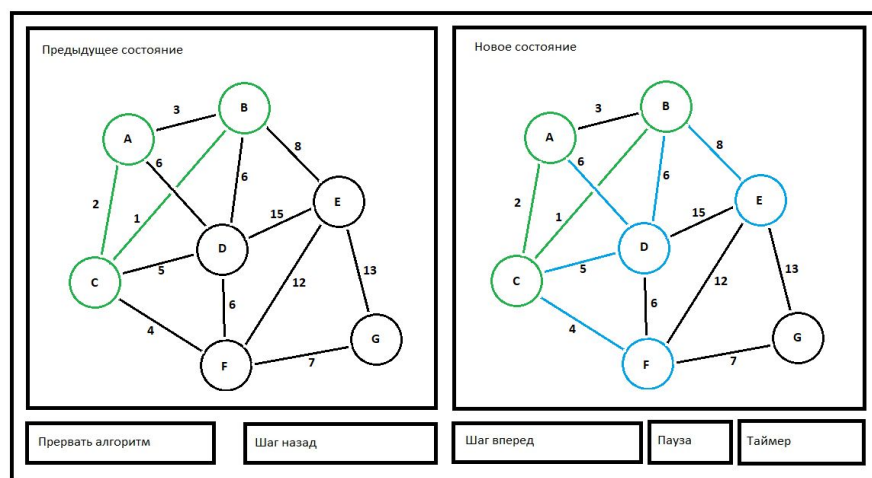
Отмена

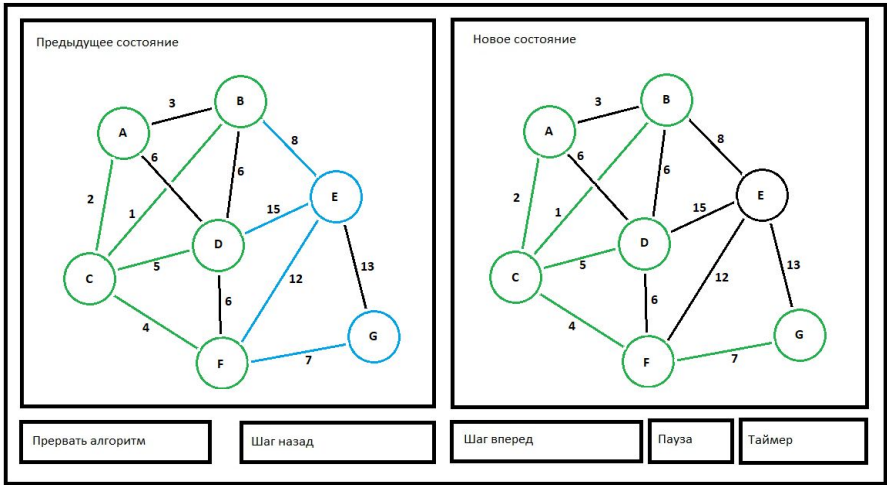
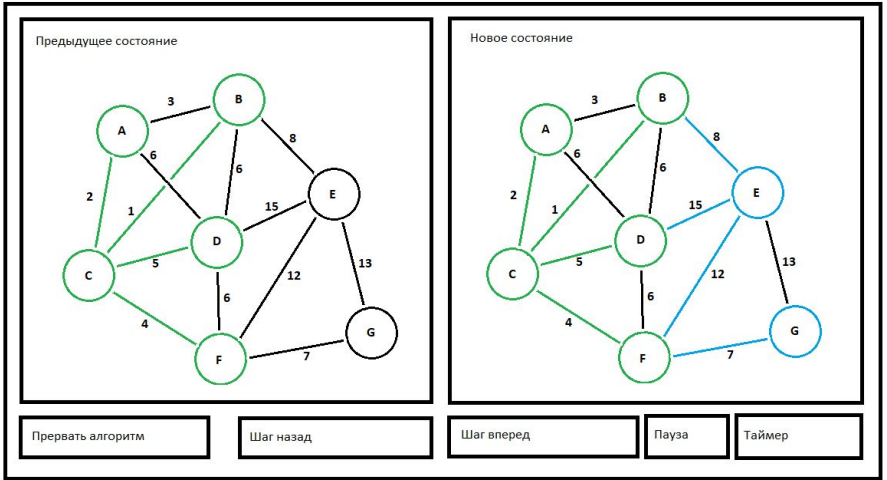
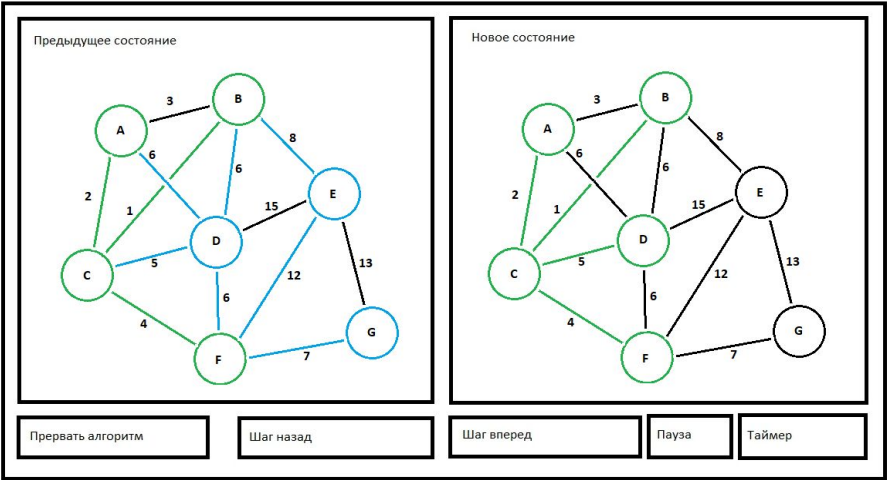
Готово

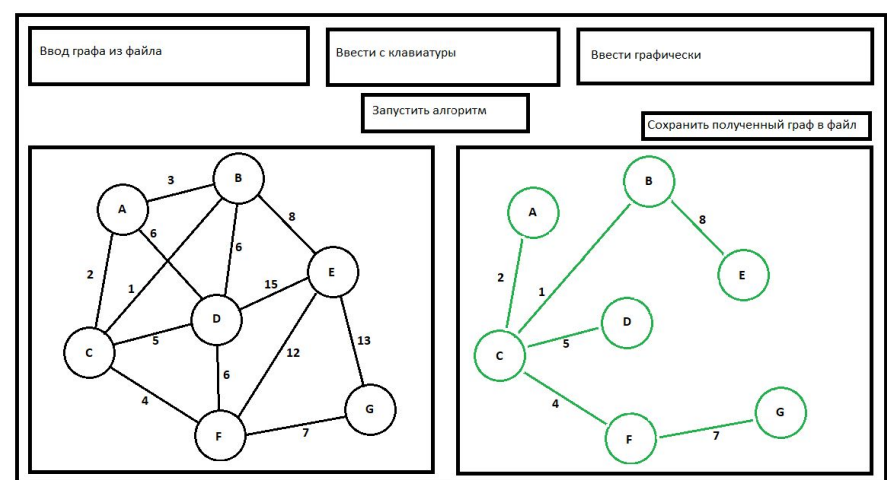
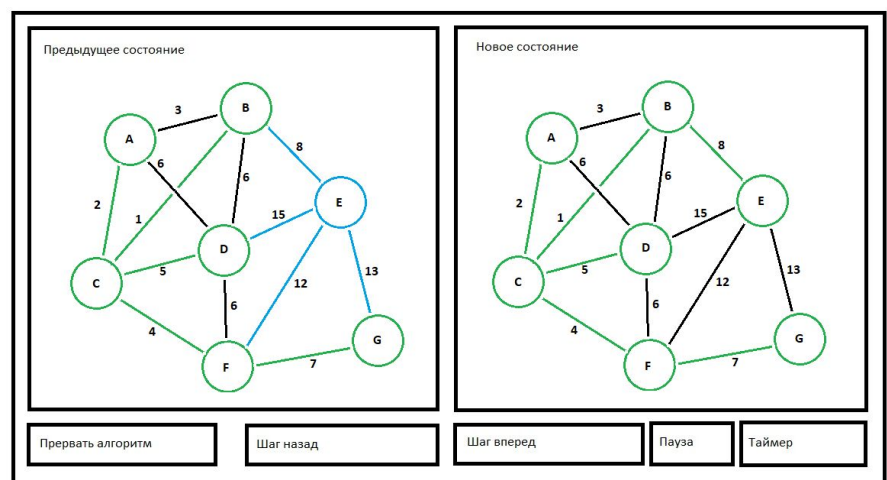
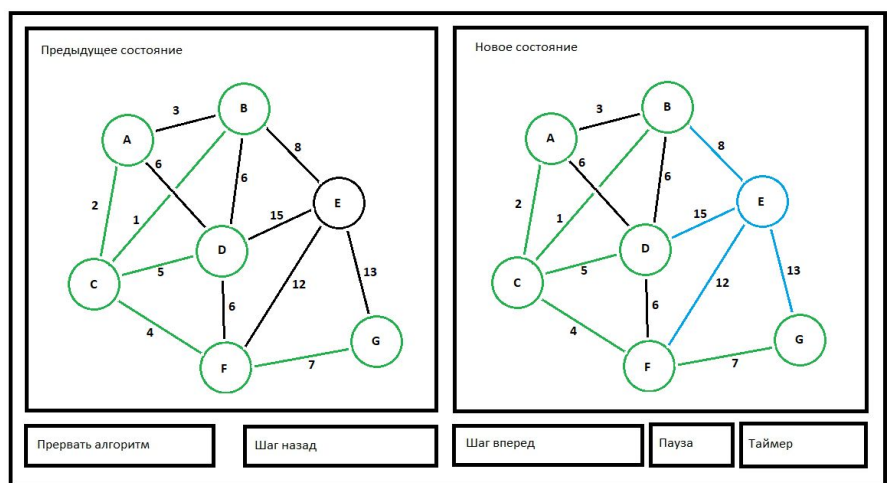
Рисунок 3 - Окно графического ввода графа

ПРИМЕР РАБОТЫ АЛГОРИТМА









ПЛАН

Прототип:

Будут реализованы все окна программы. Все кнопки интерфейса будут реализовать только открытие или закрытие соответствующих им окон, остальные кнопки будут отключены.

Итерация 1:

Будет реализован ввод графа с клавиатуры или из файла и сохранение графа в файл. Будет реализован пошаговый проход алгоритма с комментариями в соответствующей области без возможности отмены действий.

Итерация 2:

Будет реализовано отображение графа в соответствующих полях на главном окне и на окне, отображающем текущее состояние. Будет добавлен ввод графа с помощью графического интерфейса.

Итерация 3:

В программу будет добавлен автоматический переход к следующему шагу алгоритма по таймеру, также будет реализовано изменение времени таймера и его отключение. Будет реализована функция возврата к предыдущему шагу алгоритма (до 32 шагов).

КОМАНДНЫЕ РОЛИ

Сычевский Радимир – реализация связи между алгоритмом и интерфейсом, написание отчета

Иолшина Валерия – реализация графического интерфейса, тестирование графического интерфейса

Власов Роман – реализация алгоритма и визуализации графа, тестирование алгоритма