ТЗ. Алгоритм Прима

1. Реализовать интерактивное обучение алгоритму работы по нахождения минимального сотового дерева неориентированого связного взвешенного графа, то есть графа, соединяющего все вершины исходного, но с наименьшей суммой весов ребер (расстояний между вершинами), методом Прима.

1.1 Сайт предназначен для того, чтобы обучить пользователя пользоваться данным алгоритмом и уметь применять его на практике. Алгоритмы используются для нахождения кратчайшего остового взвешенного связного неориентированого графа.

1.2 Для функционирования алгоритма необходимо составить матрицу смежности. Далее, путем удаления строк и поисков минимальных элементов необходимо получить последовательность ребер, которая и будет являться минимальным оптовым деревом.

1.3 Математическая составляющая: работа с графами, хранение графов, обходы графов.

1.4 Для повышения наглядности необходим вывод графа, а также матрицы смежности на экран. Так же необходимо придумать такой способ хранения графа, чтобы можно было связать графическую часть данных (координаты вершин, активность ребер..) и алгоритмическую часть(матрицу смежности, вес ребра..). При выборе минимального значения среди одинаковых берется ребро с наименьшей по номеру второй координатой.

2. Обучение проводится в двух режимах:

* + Тестирование. Здесь пользователь знакомится с интерфейсом среды, изучает работу алгоритма на примере компьютера, который пошагово демонстрирует ход работы алгоритма.
  + Контроль. Здесь пользователю дается граф, а он сам рисует по алгоритму минимальное остовое , дерево. Далее его работа проверяется и выдается результат об успешном или неуспешном прохождении теста

Функционал

* + Генерация графа. Точнее генерация весов ребер. Вершины остаются статичными. Возможен маленький разброс в количестве ребер, тк есть вероятность, что ребро окажется с нулевым весом, тогда, в условиях программы оно не существует. Важный момент в том, что генерируется сперва матрица смежности, а потом создается структура для хранения графа.
  + Структура для хранения графа. Для удобства реализации программы была разработана следующая структура данных:

mas[i]: object{

q1:object{ Первая вершина ребра

num, Номер вершины

x, Координата вершины по оси х

y Координата вершины по оси y

},

q2:object{ Вторая вершина ребра (аналогично первой)

num,

x,

y

},

weight, Вес ребра

vis Активность ребра

}

Таким образом, в памяти храниться массив ребер графа

* + Построение на экране компьютера графически считанного графа. Для удобства вывода графа мною использована библиотека CanvaScript. Отдельной функцией рисуются ребра, отдельной функцией рисуются вершины. Предусмотрен вывод веса ребра при наведение на него указателем мыши.
  + Функции создания матрицы смежности по структуре данных и наоборот, структуры данных по матрице. Необходимо для реализации алгоритма.
  + Построение сотового дерева по алгоритму Прима. Возвращает порядок идущих друг за другом ребер сотового дерева.
  + Проверка правильности дерева, построенного пользователем. Сравнивает масив последовательности ребер, которы получился у пользователя с массивом, который получился у компьютера в ходе выполнения алгоритма.