

## Контрольное мероприятие №4

Для выполнения КМ на «удовлетворительно», необходимо выполнить первый пункт. Для выполнения КМ на «хорошо», необходимо выполнить первые два пункта. Для выполнения КМ на «отлично», необходимо выполнить все пункты. Номер задачи соответствует номеру студента в журнале.

### 1) Графы

- 1.1) Для графа согласно варианту реализовать его представление в виде списка смежности и матрицы смежности.
- 1.2) Используя оба представления реализовать базовые функции:
  - 1.2.а) Ориентированный ли граф (считать граф простым)?
  - 1.2.б) Сколько исходящих связей у вершины, закрашенной зелёным?
  - 1.2.в) Существует ли ребро между вершиной красной и вершиной жёлтой?
  - 1.2.г) Каков вес данного ребра?
  - 1.2.д) Вывести список всех вершин, связанных с синей с указанием их веса.
- 1.3) Реализовать представление графа в NetworkX или PyVis, визуализировать граф.
- 1.4) Отобразить путь от зелёной вершины до красной, перекрасив соответствующие связи. Путь выбрать самостоятельно любой из возможных.

### 2) Поиск пути по графу

- 2.1) Написать собственный алгоритм поиска пути в ширину. Проверить алгоритм на лабиринте согласно индивидуальному заданию.
- 2.2) Написать собственный алгоритм поиска пути в глубину. Проверить алгоритм на лабиринте согласно индивидуальному заданию.
- 2.3) Написать собственный алгоритм поиска пути по методу Дейкстры. Проверить алгоритм на лабиринте согласно индивидуальному заданию.
- 2.4) Для алгоритмов из пункта 1-3 провести исследование на сложность алгоритма. Сравнить с реализацией алгоритмов в SciPy и NetworkX.
- 2.5) Предложить вариант визуализации любого из алгоритмов п. 1 — 4 для отображения найденного пути на рисунке лабиринта.

### 3) Пространственные деревья

- 3.1) Дан набор точек, координаты точки разделены пробелом. Одна строка — одна точка. Выполнить поиск ближайшего соседа для первой точки. Поиск провести жадным методом и используя k-d-дерево. Сравнить k-d-дерево, построенное самостоятельно, и аналогичное дерево, построенное с использованием KDTree или cKDTree из библиотеки scipy.spatial, провести поиск соседа, используя эту библиотеку. Провести сравнение быстродействия алгоритмов для возрастающего числа точек.
- 3.2) Дан набор многоугольников. Многоугольники заданы точками - вершинами с парой координат, заданных аналогично п. 1. Вершины разделены точкой с запятой. Провести поиск пересекающихся многоугольников, используя метод Монте — Карло. Выполнить визуализацию исходной картины, стороны

многоугольников покрасить в чёрный цвет. Пересекающиеся многоугольники закрасить красным. Многоугольники, не пересекающиеся с другими многоугольниками, закрасить зелёным.