

Задания на допуск к экзамену:

1. Алгоритм: DFS/BFS: поиск компонент связности графа и слабой связности в орграфе.
2. Поиск мостов и шарниров в графе.
3. Построение остовного дерева графа: DFS или BFS.
4. Алгоритм Флойда-Уоршелла:
  - a) вектор степеней вершин,
  - b) эксцентриситеты,
  - c) диаметр, множество периферийных вершин,
  - d) радиус, множество центральных вершин.
5. Определить, является ли граф двудольным, вывести состав долей.
6. Найти проход в лабиринте от начальной точки до конечной.

Задания на 3 автоматом:

7. Определить компоненты сильной связности в орграфе.
8. Поиск мостов и шарниров в графе за линейное (по количеству рёбер) время.
9. Поиск минимального остовного дерева.
10. Алгоритм Флойда-Уоршелла:
  - a) восстановление пути для выбранной пары вершин,
  - b) характеристики каждой компоненты связности графа.
11. Алгоритм Беллмана-Форда-Мура.
12. Поиск пути на карте алгоритмом  $A^*$  с эвристиками: евклидовой, Чебышева, манхэттенской.
13. Найти максимальное паросочетание в двудольном графе.
14. Найти максимальный поток в сети алгоритмом Форда-Фалкерсона.
15. Обход всех вершин графа (алгоритм муравьиной колонии).

Задания на 4 автоматом:

16. Поиск остовного дерева алгоритмом Борувки с параллельными потоками.
17. Распределение вершин в графе по областям, в зависимости от удалённости от заданной вершины. Всего выделяется 4 области.
18. Обход всех вершин графа (метод ветвей и границ).
19. Обход всех вершин графа (алгоритм муравьиной колонии с параллельными потоками).
20. Определение планарности графа.
21. Гамма-алгоритм определения планарности графа.

Описание входных данных для графа (нулевая строка файла содержит количество вершин, вершины нумеруются с 1):

1. Граф, заданный списком рёбер. Каждое ребро хранится отдельной строкой. В строке три числа, разделённых пробелом: исходящая вершина, входящая вершина, вес ребра.
2. Граф, заданный списками смежности. Номер строки представляет собой номер вершины графа, в строке перечислены смежные вершины и инцидентные рёбра в формате: номер смежной вершины : вес инцидентного ребра.
3. Граф, заданный матрицей смежности. Значения в строках разделены пробелами.

Описание входных данных (лабиринт):

1. Двумерная матрица, где 0 – нет прохода, 1 – проход.

Описание входных данных (карта):

1. Двумерная матрица, представляющая собой строки со значениями «высот», разделёнными пробелами. Высота 0 означает, что прохода нет.

Требования к представлению данных в программах:

Для хранения графа/орграфа используется класс Graph, в котором данные закрыты и представляют собой один из двух типов данных:

1. Матрица смежности.
2. Списки смежности.

Для работы с графом используются следующие открытые функции:

- a) конструктор класса – принимает путь к файлу, в котором хранится граф и тип файла;
- b) `size` – возвращает количество вершин в графе/орграфе;
- c) `weight` – принимает номера двух вершин, возвращает вес ребра/дуги, связывающего их;
- d) `is_edge` – принимает номера двух вершин и возвращает `True`, если в графе/орграфе есть соответствующее ребро/дуга, `False`, если ребра/дуги нет;
- e) `adjacency_matrix` – возвращает матрицу смежности графа/орграфа;
- f) `adjacency_list` – принимает номер вершины, возвращает список вершин, смежных данной вершине в графе / исходящих из данной вершины в орграфе;
- g) `list_of_edges` – без аргументов возвращает список всех рёбер графа; с аргументом – номер вершины, возвращает список рёбер графа, инцидентных данной вершине / дуг, исходящих из данной вершины.
- h) `is_directed` – возвращает `True`, если граф ориентированный, `False`, если граф простой.

Для хранения карты (лабиринта) используется класс `Map`, в котором данные закрыты и представляют собой матрицу, в которой  $i$ -я строка и  $j$ -й столбец задают «клетку» на карте. Значение представляет собой высоту местности в данной точке. Нулевое значение высоты означает, что прохода нет. Переход возможен только между соседними «клетками» по вертикали или горизонтали. Расстояние между клетками выражается следующей функцией:  $h(ij, kl) = \text{abs}(k - i) + \text{abs}(l - j) + \text{abs}(a_{kl} - a_{ij})$ , где  $ij$  – координаты текущей «клетки»,  $kl$  – координаты соседней клетки,  $a_{ij}$  – значение в текущей «клетке», обозначающее высоту в этой точке,  $a_{kl}$  – значение высоты в «клетке» с координатами  $kl$ . Эвристики задаются следующими функциями ( $sp$  – координаты конечной точки маршрута):

- a. Манхэттенское расстояние  $d(ij, sp) = \text{abs}(s - i) + \text{abs}(p - j)$ .
- b. Расстояние Чебышева  $d(ij, sp) = \max(\text{abs}(s - i), \text{abs}(p - j))$ .
- c. Евклидово расстояние  $d(ij, sp) = \sqrt{(s - i)^2 + (p - j)^2}$ .

Класс содержит следующие открытые функции:

- a) конструктор класса – принимает путь к файлу;
- b) индексатор – принимает номер строки и столбца, возвращает значение высоты в точке;
- c) `size` – возвращает число строк и столбцов в матрице, представляющей карту;
- d) `neighbors` – принимает номер строки и столбца, возвращает список соседних клеток.