

# Базовая часть

1. Реализовать алгоритм Беллмана-Форда и алгоритм Джонсона.  
Граф задан в виде матрицы смежности.  
Структура файла: бинарный файл (int16), где первое число определяет размер считываемой матрицы. В программе данные хранятся в виде двумерного массива.  
Программа принимает обязательный параметр – имя входного файла. Имя выходного задаётся ключом `-o filename`, если данный ключ не указан, пишется в файл по умолчанию. Выходной файл представляет собой текстовый файл, в котором указывается:
  - а) содержит ли граф цикл с отрицательным весом;
  - б) вектор расстояний (в строку через пробел) от нулевой вершины, до всех остальных, вычисленный по алгоритму Беллмана-Форда для графов, не содержащих цикл с отрицательным весом;
  - в) диаметр, радиус, множество центральных и множество периферийных вершин (определённые из матрицы расстояний, вычисленной по алгоритму Джонсона) для графов, не содержащих цикл с отрицательным весом.
2. Реализовать алгоритм Форда-Фалкерсона.  
Граф задан в виде списков смежности.  
Структура файла: бинарный файл (int16), где первое число определяет количество вершин графа. Далее следует перечисление рёбер графа, заданное тремя значениями: начало ребра, конец ребра, вес ребра. В программе данные хранятся в виде списков смежности (массив указателей на начало списка, тело узла – номер концевой вершины и вес ребра).  
Программа принимает обязательный параметр – имя входного файла. Имя выходного задаётся ключом `-o filename`, если данный ключ не указан, пишется в файл по умолчанию. Выходной файл представляет собой текстовый файл, в котором указывается:
  - а) величина потока;
  - б) величина потока и пропускная способность для каждого ребра.
3. Реализовать алгоритмы Дейкстры и A\* (с эвристиками Чебышева, Евклидовой, Манхэттенской).  
Граф задан в виде карты, представляющей собой двумерную матрицу со значениями «высот».  
Структура файла: бинарный файл (int16), где первые два числа определяют количество клеток по горизонтали и вертикали. Далее хранятся неотрицательные значения, обозначающую высоту в данной клетке. В программе данные хранятся в виде двумерной матрицы.

Программа принимает обязательный параметр – имя входного файла. Имя выходного задаётся ключом `-o filename`, если данный ключ не указан, пишется в файл по умолчанию. Выходной файл представляет собой текстовый файл, в котором для каждого из алгоритмов (Дейкстры + 3 эвристики) указывается:

- a) длина найденного пути,
- b) процент просмотренных вершин,
- c) время выполнения алгоритма,
- d) путь;

4. Реализовать структуру данных – рандомизированное дерево бинарного поиска.

Структура файла: бинарный файл (int32).

Программа, демонстрирующая работу с деревом, принимает обязательный параметр – имя входного файла. Далее пользователю предоставляется интерфейс с операциями поиска элемента, вставки элемента, вывода наименьших 10 элементов. Для каждой из операций указывать время выполнения.

## Дополнительная часть

1. Реализовать поиск минимального остовного дерева в графе посредством алгоритма Борувки. на параллельных потоках.

Граф задан в виде списков смежности.

Структура файла: бинарный файл (int16), где первое число определяет количество вершин графа. Далее следует перечисление рёбер графа, заданное тремя значениями: начало ребра, конец ребра, вес ребра. В программе данные хранятся в виде списков смежности (массив указателей на начало списка, тело узла – номер концевой вершины и вес ребра).

Программа принимает обязательный параметр – имя входного файла. Имя выходного задаётся ключом `-o filename`, если данный ключ не указан, пишется в файл по умолчанию. Выходной файл представляет собой текстовый файл, в котором указывается:

- a) суммарный вес рёбер минимального остовного дерева;
- b) ребра, входящие в минимальное остовное дерево.

2. Реализовать алгоритм поиска паросочетаний.

Граф задан в виде матрицы смежности.

Структура файла: бинарный файл (int16), где первое число определяет размер считываемой матрицы. В программе данные хранятся в виде двумерного массива.

Программа принимает обязательный параметр – имя входного файла. Имя выходного задаётся ключом `-o filename`, если данный ключ не указан, пишется в файл по умолчанию. Выходной файл представляет собой текстовый файл, в котором указано:

- a) является ли граф двудольным;
- b) число паросочетаний (для двудольного графа);
- c) ребра, входящие в паросочетание (для двудольного графа).

3. Реализовать поиск компонент связности и двусвязности в графе.

Граф задан в виде матрицы смежности.

Структура файла: бинарный файл (int16), где первое число определяет размер считываемой матрицы. В программе данные хранятся в виде двумерного массива.

Программа принимает обязательный параметр – имя входного файла. Имя выходного задаётся ключом `-o filename`, если данный ключ не указан, пишется в файл по умолчанию. Выходной файл представляет собой текстовый файл, в котором перечисляются:

- a) мосты и точки сочленения;
- b) компоненты двусвязности;
- c) компоненты связности.

4. Реализовать структуру данных – красно-чёрное дерево.

Структура файла: бинарный файл (int32).

Программа, демонстрирующая работу с деревом, принимает обязательный параметр – имя входного файла. Далее пользователю предоставляется интерфейс с операциями поиска элемента, вставки элемента, вывода наименьших 10 элементов. Для каждой из операций указывать время выполнения.