

## Лабораторная работа №8

### Задание на работу.

1. Написать функцию `depth-first`, которая для заданного взвешенного ориентированного графа с петлями находит путь наименьшей стоимости между двумя его произвольными вершинами с использованием поиска в глубину.
2. Написать функцию `breadth-first`, которая для заданного взвешенного ориентированного графа с петлями находит путь наименьшей стоимости между двумя его произвольными вершинами с использованием поиска в ширину.
3. Написать функции
  - a) `degree(Graph Node)` определяющую степень заданной вершины графа.
  - b) `nodes-list(Graph)` генерирующую список всех вершин графа, отсортированных в порядке убывания их степеней.

### Формы задания графа.

#### 1. *Arc-clause form*

```
( (a b 7) (b c 9) (c a 5) ;; loop
(k k 0) ) ;; standalone vertex
```

Т.е., граф задается перечислением его взвешенных дуг.

#### 2. *Graph-term form*

```
((k m p q) (m p 7) (p m 5) (p q 9))
% k - standalone vertex
```

Первый список задает вершины графа. Последующие списки вида `(m p 7)` определяют дугу из `m` в `p` с весом 7.

#### 3. *Adjacency-list form*

```
( (k () (0)) (m (q) (7)) (p (m q) (5 9)) )
% k - standalone vertex
```

Граф есть список списков вида `(p (m q) (5 9))`, где `p` – вершина графа, `(m q)` – список вершин, в которые есть дуги из `p`, `(5 9)` – список весов соответствующих дуг.

Способ представления графа определяется номером зачетной книжки, взятым по модулю 3:

0 – *Arc-clause form*

1 – *Graph-term form*

2 – *Adjacency-list form*