

**Тема:** Компоненты связности (найти количество компонент связности графа, вывести компоненты списком вершин согласно нумерации в матрице смежности).

Для начала рассмотрим алгоритм *DFS* поиска в глубину, на котором базируется алгоритм решения задачи. Существует множество вариантов алгоритма *DFS*, в зависимости от задачи: формирование леса поиска по исходному графу, расстановка меток времени и т.д. В данном случае нам нужно в процессе обхода графа, начав с определенной вершины, запомнить все вершины, до которых дойдет алгоритм, тем самым будет сформирована компонента связности этой вершины. *DFS* использует стек для хранения вершин в процессе обхода, в данном случае будем использовать стек вызовов для этой цели, то есть алгоритм *DFS* будет рекурсивным. Таким образом, функцию *DFS* можно представить в виде следующих шагов:

$DFS(v, visited, matrix)$

{**In:** вершина  $v \in V(G)$ ; множество *visited* посещенных вершин (при первом запуске пустое); неориентированный граф  $G$ , заданный матрицей смежности *matrix*}

{**Out:** список всех вершин, до которых дошел алгоритм в процессе обхода}

- (1) Добавляем вершину  $v$  в множество *visited*;
- (2) С помощью матрицы смежности обходим все вершины, смежные с  $v$ ;
- (3) Для очередной вершины  $u$ , смежной с  $v$  и не лежащей в множестве *visited*, запускаем:  $visited = DFS(u, visited, matrix)$ ;
- (4) Возвращаем множество *visited* всех посещенных в процессе обхода вершин.

**Описание алгоритма:** Пусть  $Q$  - множество всех вершин графа  $G$  и *result* - количество компонент связности (инициализируется нулем). Далее:

- (1) Извлекаем из  $Q$  вершину  $v = Q[0]$ . Запускаем  $component = DFS(v, \{\}, matrix)$ . Функция *DFS* полностью определяет компоненту связности *component*, в которой находится вершина  $v$ ;
- (2) *component* выводится на экран;
- (3) Из  $Q$  удаляются все вершины, которые находятся в *component*;
- (4) Инкрементируем переменную *result*;
- (5) Если  $Q$  не пусто, то возвращаемся к шагу (1). В противном случае выводим *result* и завершаем работу.