

# DFS (обход графа в глубину)

Вайцуль А. Н.

## 1 Постановка задачи

Поиск в глубину — это рекурсивный алгоритм, использующий концепцию обратного отслеживания. Он включает в себя тщательный поиск всех узлов, продвигаясь вперед, если это возможно, или возвращаясь назад. Здесь слово «обратный путь» означает, что, как только вы двигаетесь вперед и на текущем пути больше нет узлов, вы продвигаетесь назад по эквивалентному пути, чтобы найти узлы, которые нужно пройти. Все узлы на текущем пути будут посещены до тех пор, пока не будут пройдены все непосещенные узлы, после чего будут выбраны последующие пути.

## 2 Описание алгоритма

### Общая идея

Общая идея алгоритма состоит в следующем: для каждой не пройденной вершины необходимо найти все не пройденные смежные вершины и повторить поиск для них.

### Пошаговое представление

1. Выбираем любую вершину из еще не пройденных, обозначим ее как  $u$ .
2. Запускаем процедуру  $dfs(u)$ 
  - Помечаем вершину  $u$  как пройденную.
  - Для каждой не пройденной смежной с  $u$  вершиной (назовем ее  $v$  запускаем  $dfs(v)$
3. Повторяем шаги 1 и 2, пока все вершины не окажутся пройденными.

### Реализация

```

function doDfs(G[n]: Graph): // функция принимает граф G
с количеством вершин n и выполняет обход в глубину во всем графе
    visited = array[n, false] // создаём массив посещённых
    вершины длины n, заполненный false изначально

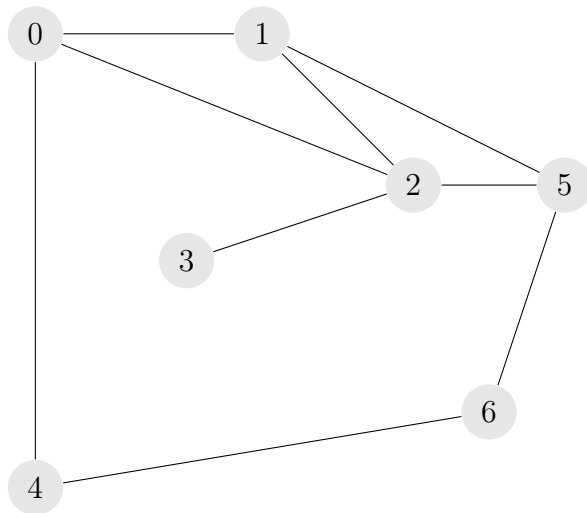
    function dfs(u: int):
        visited[u] = true
        for v: (u, v) in G
            if not visited[v]
                dfs(v)

    for i = 1 to n
        if not visited[i]
            dfs(i)

```

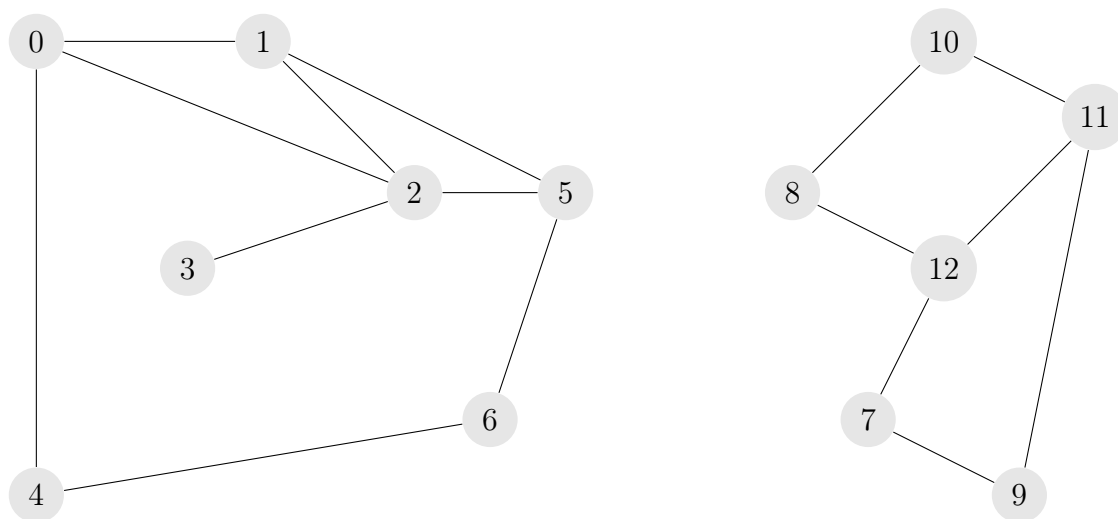
### 3 Результаты

Дан связный граф:



На выход получим последовательность обхода вершин: 2 0 4 6 5 1 3.

В случае нескольких компонент связности:



На выход получим последовательность обхода вершин: 2 5 1 0 4 6 3 и 7 12 11 10 8 9.