

Тема 6. Алгоритмы на графах

6.2. Поиск в глубину

Поиск в глубину (англ. Depth First Search) является одним из методов систематического исследования всех вершин и ребер графа. Он исходит из процедуры прохождения графа методом поиска с возвратом и является основой многих алгоритмов на графах как ориентированных, так и неориентированных.

Общая идея метода состоит в следующем. Поиск начинается с посещения некоторой вершины $v_0 \in V$. Пусть v – последняя посещенная вершина. Выбирается произвольное ребро (v, w) , инцидентное v . Если вершина w уже пройдена (посещалась ранее), осуществляется возврат в вершину v и выбирается другое ребро. Если вершина w еще не пройдена, то она посещается и процесс применяется рекурсивно к вершине w . Если все ребра, инцидентные вершине v , уже исследованы, осуществляется возврат назад по ребру (u, v) , по которому пришли в вершину v , и продолжается исследование ребер, инцидентных u . Процесс заканчивается, когда делается попытка вернуться назад из вершины v_0 , с которой начиналось исследование.

Рассмотренная процедура представлена алгоритмом 6.1. Данный алгоритм осуществляет поиск в глубину в произвольном, необязательно связном графе, заданном структурой смежности. Каждой вершине $v \in V$ графа поставлен в соответствие элемент $new(v)$, чтобы отличить уже пройденные вершины графа ($new(v) = \mathbf{false}$) от еще не пройденных ($new(v) = \mathbf{true}$).

```

for  $x \in V$  do  $new(x) \leftarrow \mathbf{true}$  // инициализация
for  $x \in V$  do if  $new(x)$  then  $DFS(x)$ 

```

```

procedure  $DFS(v)$ 
   $visit(v)$  // посещение вершины  $v$ 
   $new(v) \leftarrow \mathbf{false}$ 
  for  $w \in Adj(v)$  do if  $new(w)$  then  $DFS(w)$ 
return

```

Алгоритм 6.1. Поиск в глубину

Для неориентированного графа, представленного на рис. 6.2, заданного структурой смежности, поиск в глубину дает следующий порядок посещения вершин: a, b, c, d, e, g, f . Очевидно, что порядок прохождения вершин графа не единственен, так как ребра, инцидентные вершине, могут выбираться для рассмотрения в произвольном порядке.

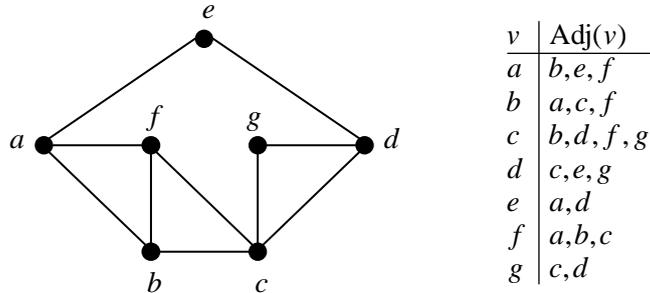


Рис. 6.2. Неориентированный граф и его структура смежности

Поскольку для каждой вершины, которая проходится впервые, производится обращение к *DFS* ровно один раз, всего требуется $|V|$ обращений к *DFS*. При каждом обращении количество производимых действий пропорционально числу ребер, инцидентных рассматриваемой вершине. Поэтому время поиска в глубину в произвольном графе равно $O(|V| + |E|)$.

Методика поиска в глубину очевидным образом переносится на ориентированные графы и не требует никаких модификаций. Необходимо только учитывать направленность ребер (дуг).

6.3. Поиск в ширину

Поиск в ширину (англ. Breadth First Search) является другим методом систематического исследования всех вершин и ребер графа. В отличие от поиска в глубину, где для реализации процесса используется стек (неявно в рекурсивном алгоритме и явно в нерекурсивном), при поиске в ширину используется очередь. Процесс поиска начинается в произвольно выбранной вершине v графа, проходятся все ребра, инцидентные v . Пусть эти ребра будут $(v, a_1), (v, a_2), \dots, (v, a_k)$. Затем исследуются ребра, инцидентные a_1, a_2, \dots, a_k . Пусть эти ребра будут $(a_1, a_{1,1}), (a_1, a_{1,2}), \dots, (a_1, a_{1,m}), (a_2, a_{2,1}), (a_2, a_{2,2}), \dots, (a_2, a_{2,n}), \dots, (a_k, a_{k,t})$. Затем исследуются ребра, инцидентные $a_{1,1}, a_{1,2}, \dots, a_{k,t}$ и т. д. Этот процесс продолжается до тех пор, пока не будут исследованы все ребра. Рассмотренная процедура поиска в ширину представлена алгоритмом 6.2.

```

for  $x \in V$  do  $new(x) \leftarrow \mathbf{true}$  // инициализация
for  $x \in V$  do if  $new(x)$  then  $BFS(x)$ 

```

```

procedure  $BFS(v)$ 

```

```

   $Q \leftarrow \emptyset$  // очередь  $Q$  пуста

```

```

   $Q \leftarrow v$ 

```

```

   $new(v) \leftarrow \mathbf{false}$ 

```

```

  while  $Q \neq \emptyset$  do
    for  $w \in Adj(v)$  do
      if  $new(w)$  then

```

$$\left\{ \begin{array}{l} v \leftarrow Q \\ \text{visit}(v) \quad // \text{посещение вершины } v \\ \left\{ \begin{array}{l} Q \leftarrow w \\ new(w) \leftarrow \mathbf{false} \end{array} \right. \end{array} \right.$$

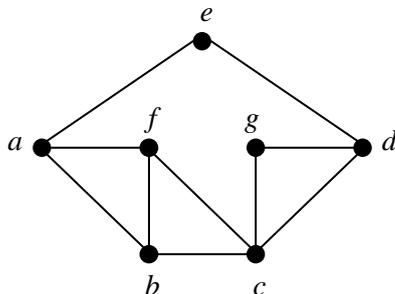
```

return

```

Алгоритм 6.2. Поиск в ширину

Данный алгоритм осуществляет поиск в ширину в произвольном, необязательно связном графе, заданном структурой смежности. Каждой вершине $v \in V$ графа поставлен в соответствие элемент $new(v)$, чтобы отличить уже пройденные вершины от еще не пройденных. Для неориентированного графа (рис. 6.2) поиск в ширину дает следующий порядок посещения вершин: a, b, e, f, c, d, g . Как и при поиске в глубину, порядок посещения вершин не единственен, так как инцидентные ребра могут выбираться в произвольном порядке.



v	$Adj(v)$
a	b, e, f
b	a, c, f
c	b, d, f, g
d	c, e, g
e	a, d
f	a, b, c
g	c, d

Вычислительная сложность алгоритма равна $O(|V| + |E|)$, так как каждая вершина помещается в очередь и удаляется из очереди в точности один раз, а число итераций в цикле **for** имеет порядок числа ребер графа.

Как и в случае поиска в глубину, процедуру поиска в ширину без всяких модификаций можно использовать и для ориентированных графов, учитывая направленность ребер (дуг).