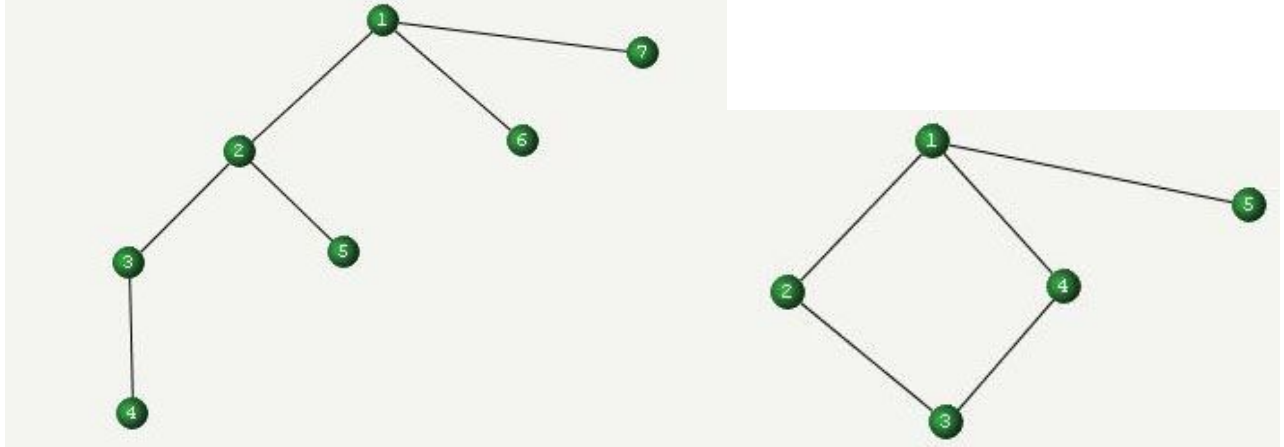


Algoritmos de Busca

- Busca em profundidade (Depth-First Search DFS)

Uma DFS começa em um vértice v chamado raiz e caminha por todos os vértices que podem ser alcançados a partir de v . Observe que v é um parâmetro passado ao algoritmo DFS. Qualquer vértice pode ser a raiz. Diremos que v é conectado a w se existir a aresta (v, w) . Os desenhos abaixo mostram a ordem de visita aos vértices de acordo com buscas em profundidade começando em v . Vértice numerado n é visitado antes de $n+1$.



Observe que, se executado manualmente por duas pessoas, o algoritmo DFS pode produzir duas numerações diferentes. Por exemplo, no grafo da esquerda acima as numerações 1 2 3 4 5 6 7 e 1 6 5 2 3 4 7 estão ambas corretas para DFS.

O algoritmo para busca em profundidade marca v (raiz) como visitado, pega um vértice w conectado a v (a aresta (v, w) existe) e continua a DFS em w . Depois que a busca em w termina, o algoritmo toma outro vértice z conectado a v ainda não visitado e faz a busca em z . O algoritmo termina quando todos os vértices ligados a v já foram marcados (visitados). O algoritmo para DFS é mostrado a seguir e incorpora código para fazer preWork e postWork, que são códigos executados ao marcar um vértice e após visitar uma aresta. PreWork e postWork dependem da finalidade para a qual DFS está sendo utilizada.

```
Algorithm DFS( $G, v$ )
Entrada:  $G = (V, E)$  e um vértice  $v \in V$ .
begin
  marque  $v$ 
  faça preWork sobre  $v$ 
  for each aresta  $(v, w) \in E$  do
    if  $w$  não foi marcado
    then
      DFS( $G, w$ );
  faça postWork para  $(v, w)$ 
end if
end
```

O algoritmo DFS pode ser usado para percorrer todos os vértices de um grafo e para descobrir se um grafo é conectado ou não. Se, após uma DFS, todos os vértices forem visitados, então o grafo é conectado.

- Busca em largura (Breadth-First Search)

Uma BFS é feita em um grafo $G = (V, E)$ começando em um vértice v . Primeiro o algoritmo visita v e todos os vértices conectados a v , chamados filhos de v . Isto é, o algoritmo visita vértices w tal que $(v, w) \in E$.

No segundo passo, o algoritmo visita todos os netos de v . Isto é, os vértices que não estão conectados diretamente a v mas estão conectados a algum vértice que está conectado a v . O algoritmo prossegue deste modo até que todos os vértices alcançáveis por v sejam visitados.

O primeiro passo do BFS visita todos os vértices que estão a uma aresta de distância de v , o segundo passo visita vértices que estão a duas arestas de distância de v e assim por diante.

O algoritmo é dado abaixo.

Algorithm BFS (G, V)

Entrada: Grafo $G = (V, E)$ e vértice v .

begin

8

 marque v

 coloque v no fim da fila F

 while F não é vazia do

 begin

 remova o primeiro vértice w de F

 faça preWork sobre v

 for each aresta (w, z) , tal que z não é marcado, do

 begin

 marque z

 insira z no fim da fila F

 end

 end

end

Não há postWork para busca em largura.