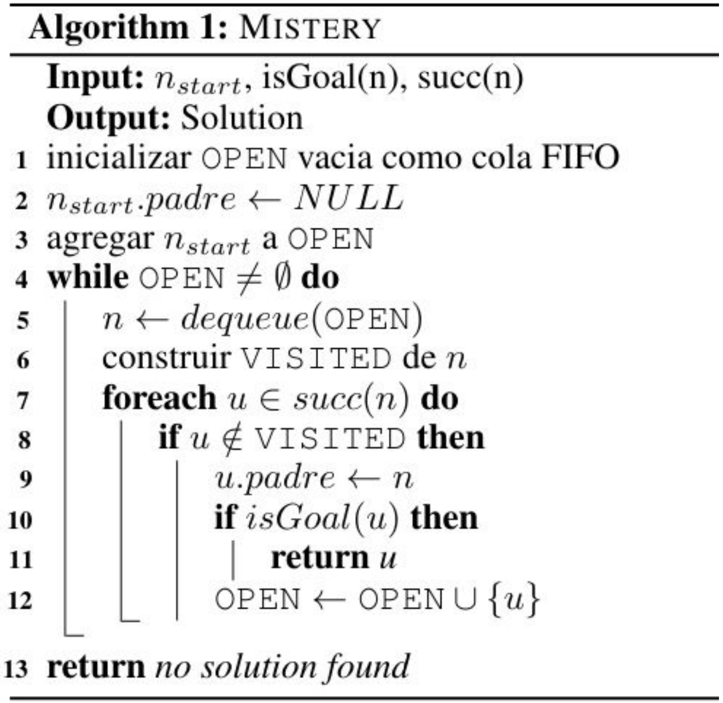
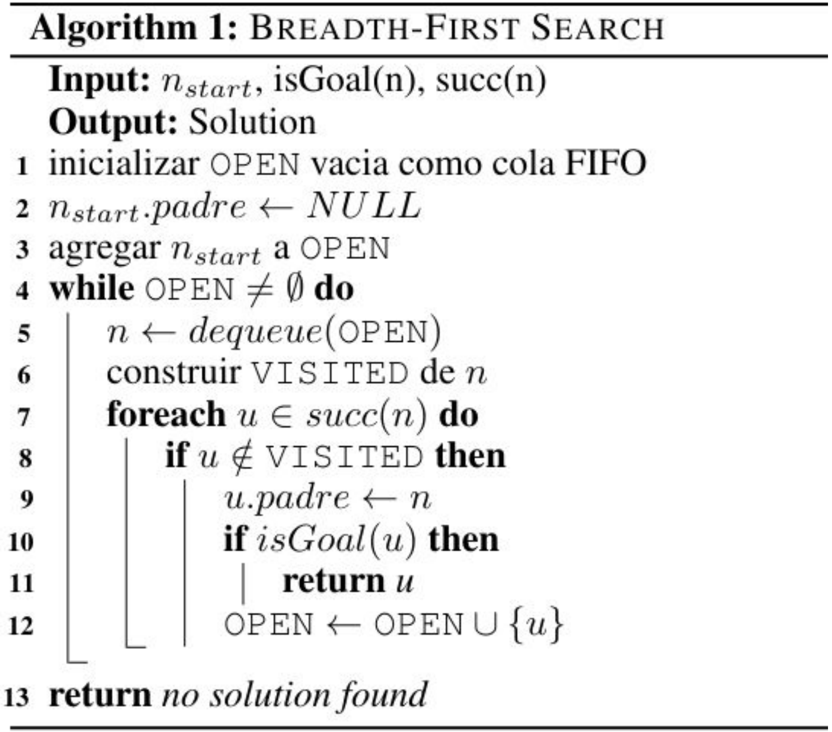
Búsqueda, recorridos y algo en común

Conocemos este algoritmo?



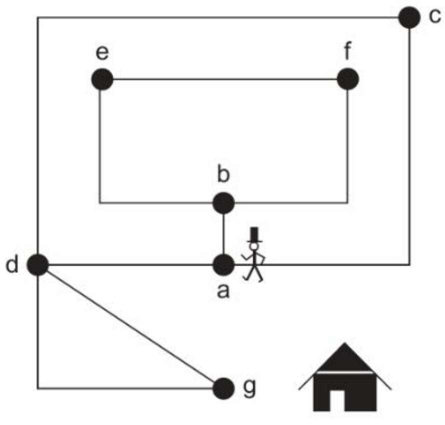
```
Algorithm 1: MISTERY
Input: $n_{start}$, isGoal(n), succ(n)
Output: Solution
inicializar OPEN vacia como cola FIFO
$n_{start}.padre \leftarrow NULL$
agregar $n_{start}$ a OPEN
while OPEN $\neq \emptyset$ do
    $n \leftarrow dequeue($OPEN$)$
    construir VISITED de $n$
    foreach $u \in succ(n)$ do
        if $u \notin$ VISITED then
            $u.padre \leftarrow n$
            if $isGoal(u)$ then
                return $u$
            OPEN $\leftarrow$ OPEN $\cup \{u\}$
return no solution found
```

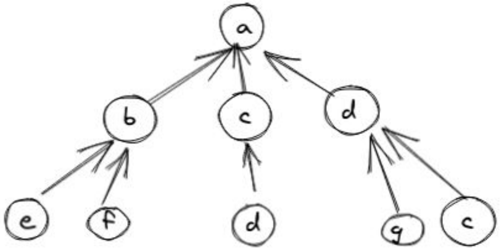
Búsqueda en anchura



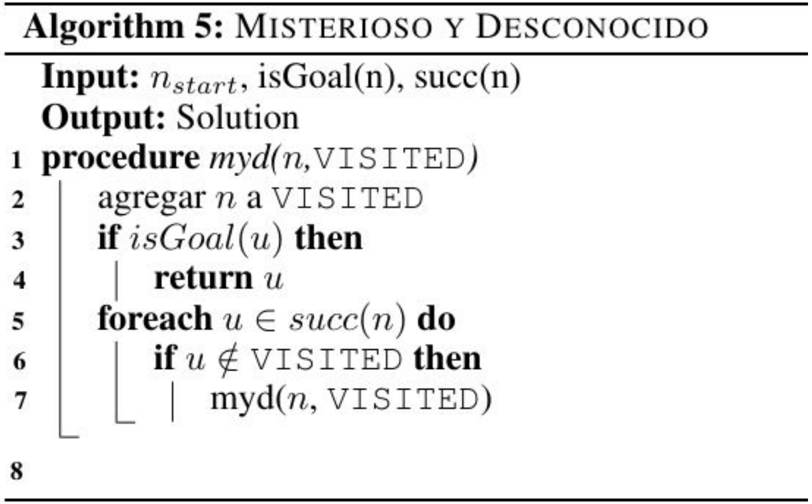
Algorithm 1: BREADTH-FIRST SEARCH

```
Input: $n_{start}$, isGoal(n), succ(n)
Output: Solution
inicializar OPEN vacia como cola FIFO
$n_{start}.padre \leftarrow NULL$
agregar $n_{start}$ a OPEN
while OPEN $\neq \emptyset$ do
    $n \leftarrow dequeue($OPEN$)$
    construir VISITED de $n$
    foreach $u \in succ(n)$ do
        if $u \notin$ VISITED then
            $u.padre \leftarrow n$
            if $isGoal(u)$ then
                return $u$
            OPEN $\leftarrow$ OPEN $\cup \{u\}$
return no solution found
```





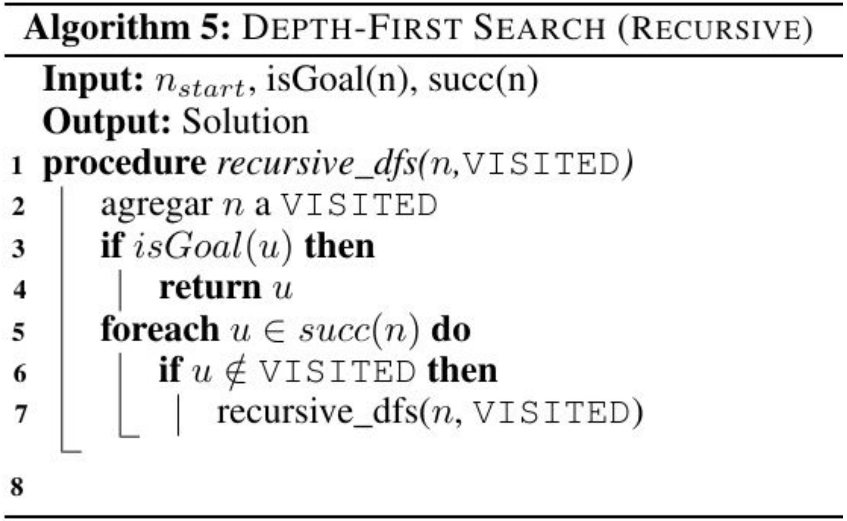
Conocemos este algoritmo?



Algorithm 5: Misterioso y Desconocido

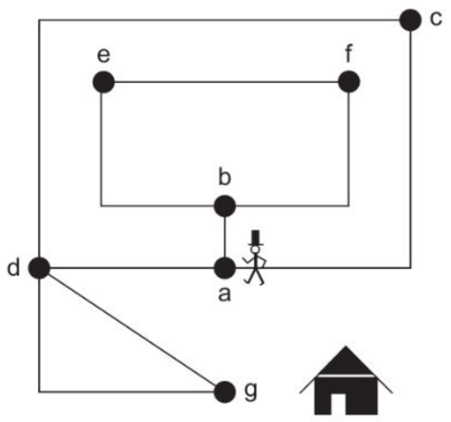
```
Input: $n_{start}$, isGoal(n), succ(n)
Output: Solution
procedure myd($n$,VISITED)
    agregar $n$ a VISITED
    if $isGoal(u)$ then
        return $u$
    foreach $u \in succ(n)$ do
        if $u \notin$ VISITED then
            myd($n$, VISITED)
```

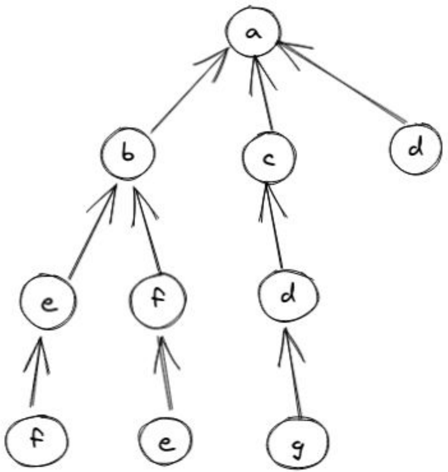
Conocemos este algoritmo?



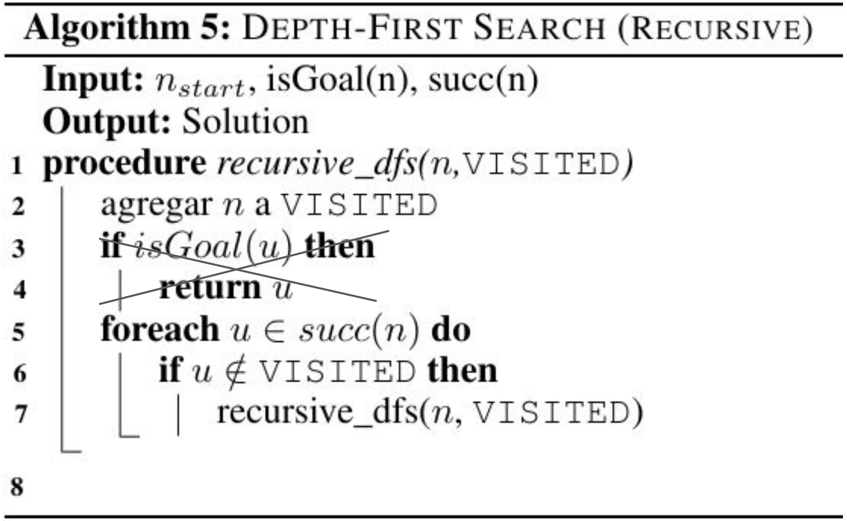
Algorithm 5: DEPTH-FIRST SEARCH (RECURSIVE)

```
Input: $n_{start}$, isGoal(n), succ(n)
Output: Solution
procedure recursive_dfs($n$,VISITED)
    agregar $n$ a VISITED
    if $isGoal(u)$ then
        return $u$
    foreach $u \in succ(n)$ do
        if $u \notin$ VISITED then
            recursive_dfs($n$, VISITED)
```





Que pasa si… ?

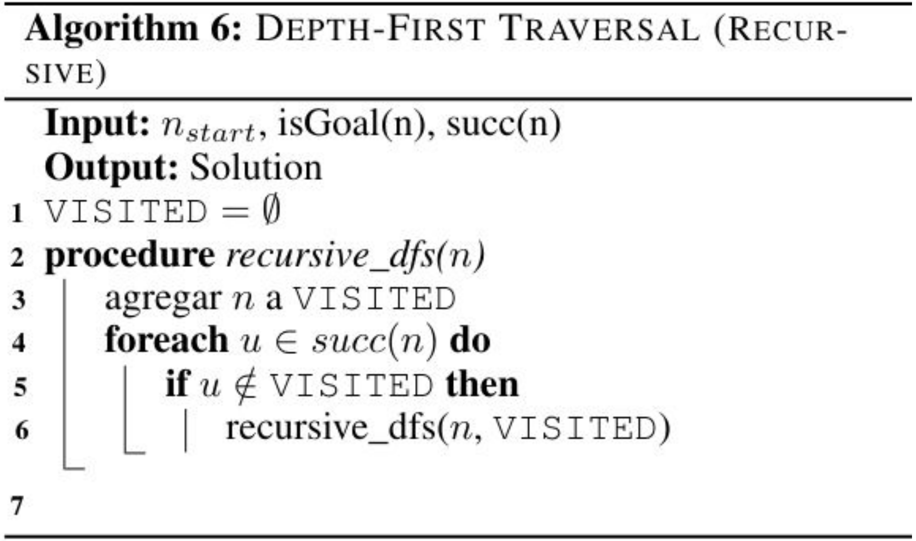


```
Algorithm 5: DEPTH-FIRST SEARCH (RECURSIVE)
Input: $n_{start}$, isGoal(n), succ(n)
Output: Solution
procedure recursive_dfs($n$,VISITED)
    agregar $n$ a VISITED
    ~~if $isGoal(u)$ then~~
        ~~return $u$~~
    foreach $u \in succ(n)$ do
        if $u \notin$ VISITED then
            recursive_dfs($n$, VISITED)
```

Que algoritmo se obtiene, si:

- No verificamos el goal
- VISITED considera los nodos del grafo en cualquier rama del espacio de estados.

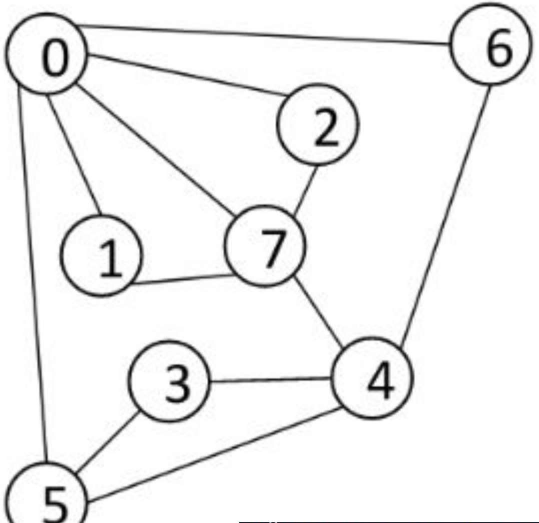
Depth First Traversal (recorrido en profundidad)

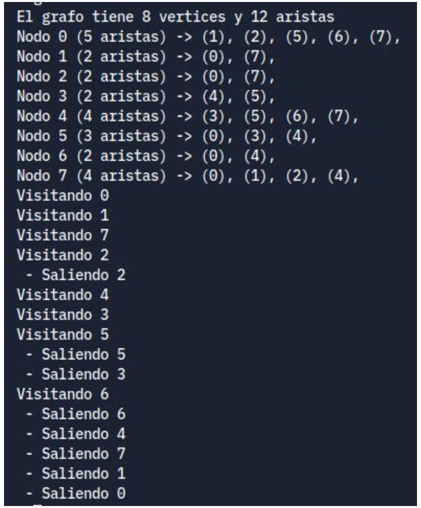


Algorithm 6: DEPTH-FIRST TRAVERSAL (RECURSIVE)

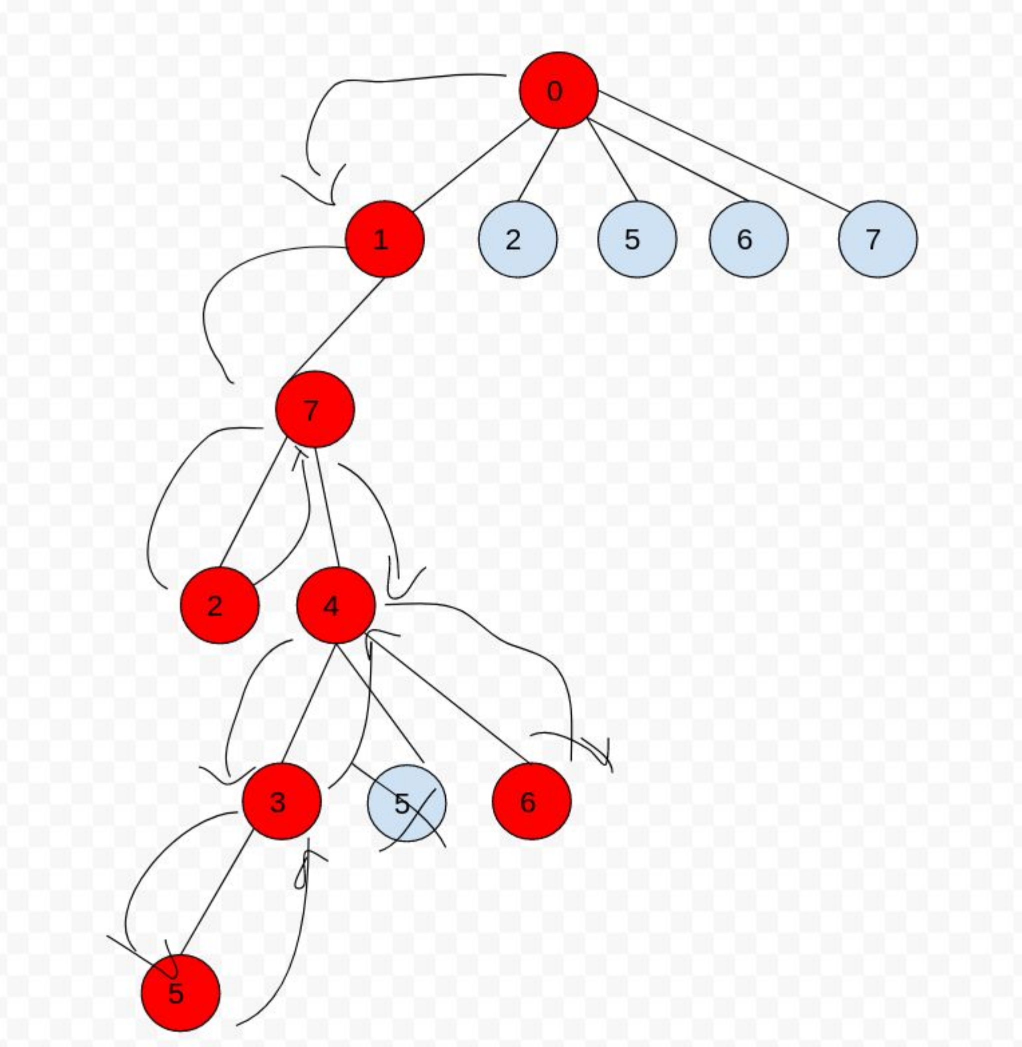
```
Input: $n_{start}$, isGoal(n), succ(n)
Output: Solution
VISITED $= \emptyset$
procedure recursive_dfs($n$)
    agregar $n$ a VISITED
    foreach $u \in succ(n)$ do
        if $u \notin$ VISITED then
            recursive_dfs($n$, VISITED)

```





```
El grafo tiene 8 vertices y 12 aristas
Nodo 0 (5 aristas) -> (1), (2), (5), (6), (7),
Nodo 1 (2 aristas) -> (0), (7),
Nodo 2 (2 aristas) -> (0), (7),
Nodo 3 (2 aristas) -> (4), (5),
Nodo 4 (4 aristas) -> (3), (5), (6), (7),
Nodo 5 (3 aristas) -> (0), (3), (4),
Nodo 6 (2 aristas) -> (0), (4),
Nodo 7 (4 aristas) -> (0), (1), (2), (4),
Visitando 0
Visitando 1
Visitando 7
Visitando 2
 - Saliendo 2
Visitando 4
Visitando 3
Visitando 5
 - Saliendo 5
 - Saliendo 3
Visitando 6
 - Saliendo 6
 - Saliendo 4
 - Saliendo 7
 - Saliendo 1
 - Saliendo 0
```


0
1
2
5
6
7
7
2
4
3
5
6
5