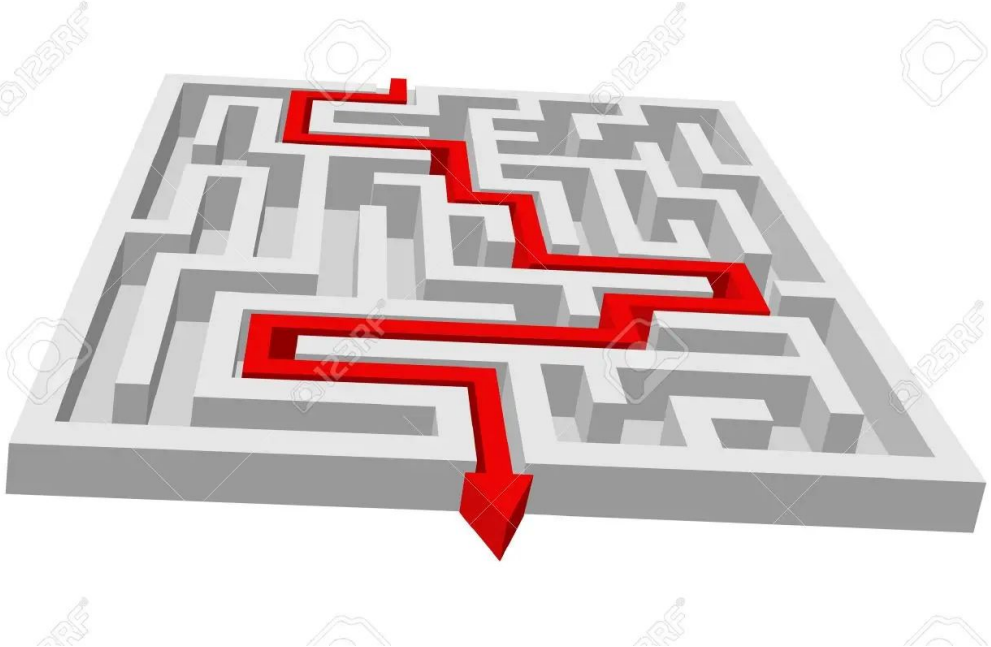
Ayudantía 4: Introducción a la Búsqueda

Benjamín Pizarro - Jaime Moreno

Contenidos

1. Qué es Buscar
2. Definición Formal
3. Ejemplo: Puzzle 8
4. Algoritmos
5. Problema Adicional

¿Qué es Buscar?



Y formalmente...

- Estado (s): Configuración específica de un sistema.
- Acción (a): Función que hace pasar al sistema de un estado a otro.
- Conjunto de acciones (A): Todas las acciones posibles.
- Espacio de Búsqueda (S): Conjunto de todos los estados posibles.
- Grafo de Búsqueda: Todos los estados posibles conectados por las acciones que los unen.

¿Cómo se define un problema de búsqueda?

- G es un subconjunto de S con los estados objetivo.
- s_init el estado inicial.
- Un problema de búsqueda es…

$$(\mathbf{S}, A, s_{init}, G)$$

Un caso clásico: El puzzle de 8

Problema de Búsqueda (S, A, s _init, G)

- $\mathcal{S}$ = conjunto de estados
- $\mathcal{A}$ = conjunto de acciones
- s_{init} = estado inicial
- G = conjunto de estados finales

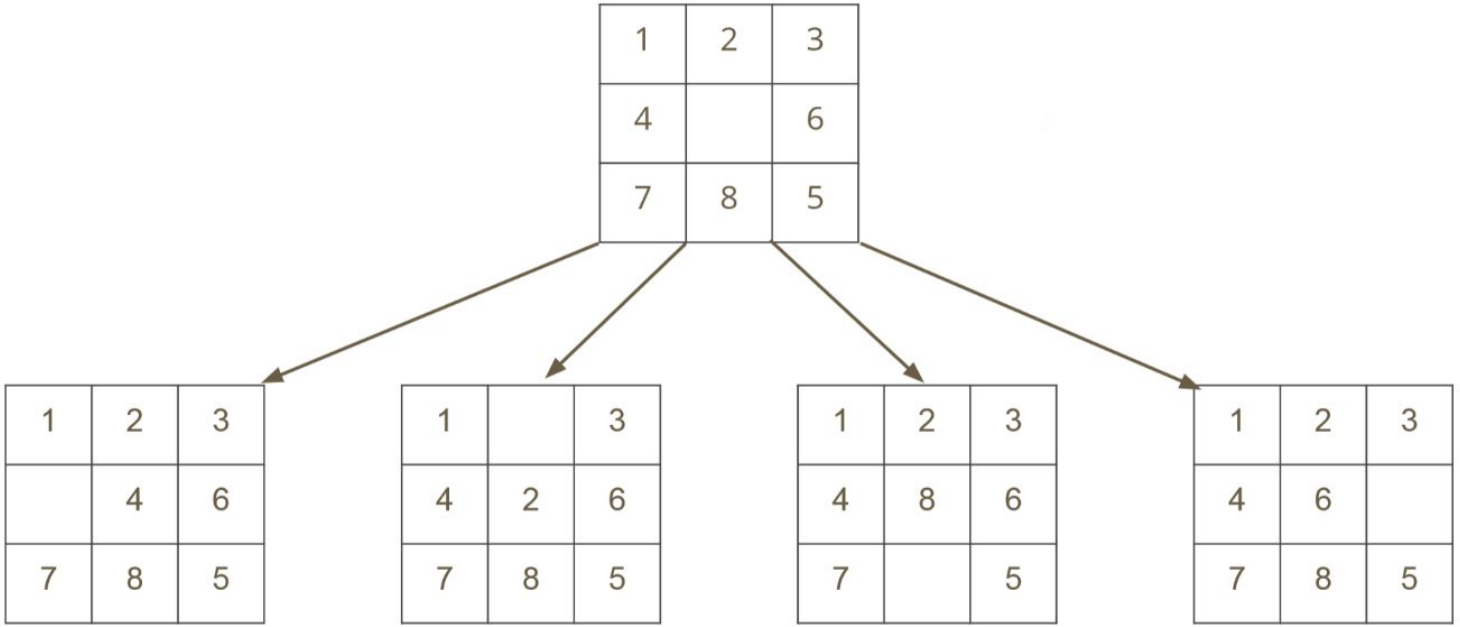
1	2	3
4		6
7	8	5

Estado inicial

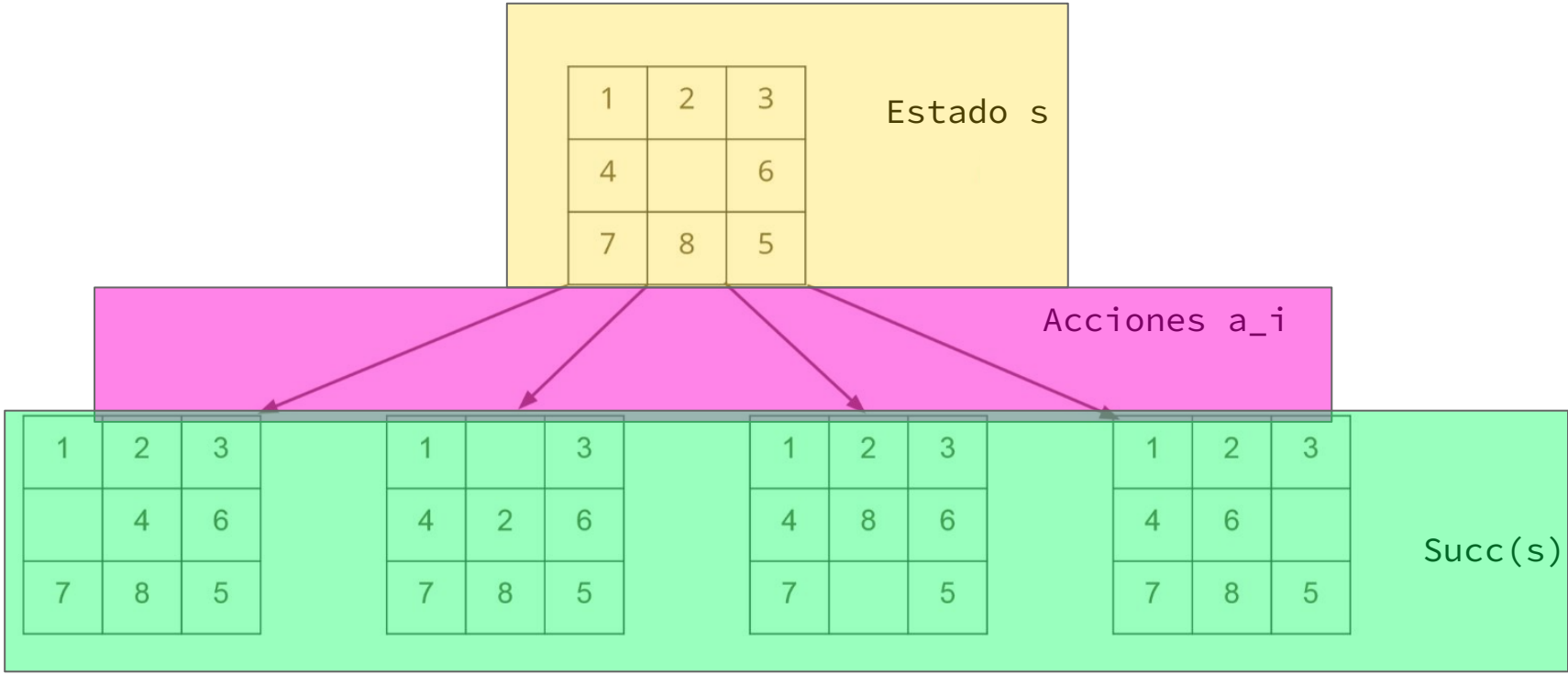
	1	2
3	4	5
6	7	8

Estado final $\in$ G

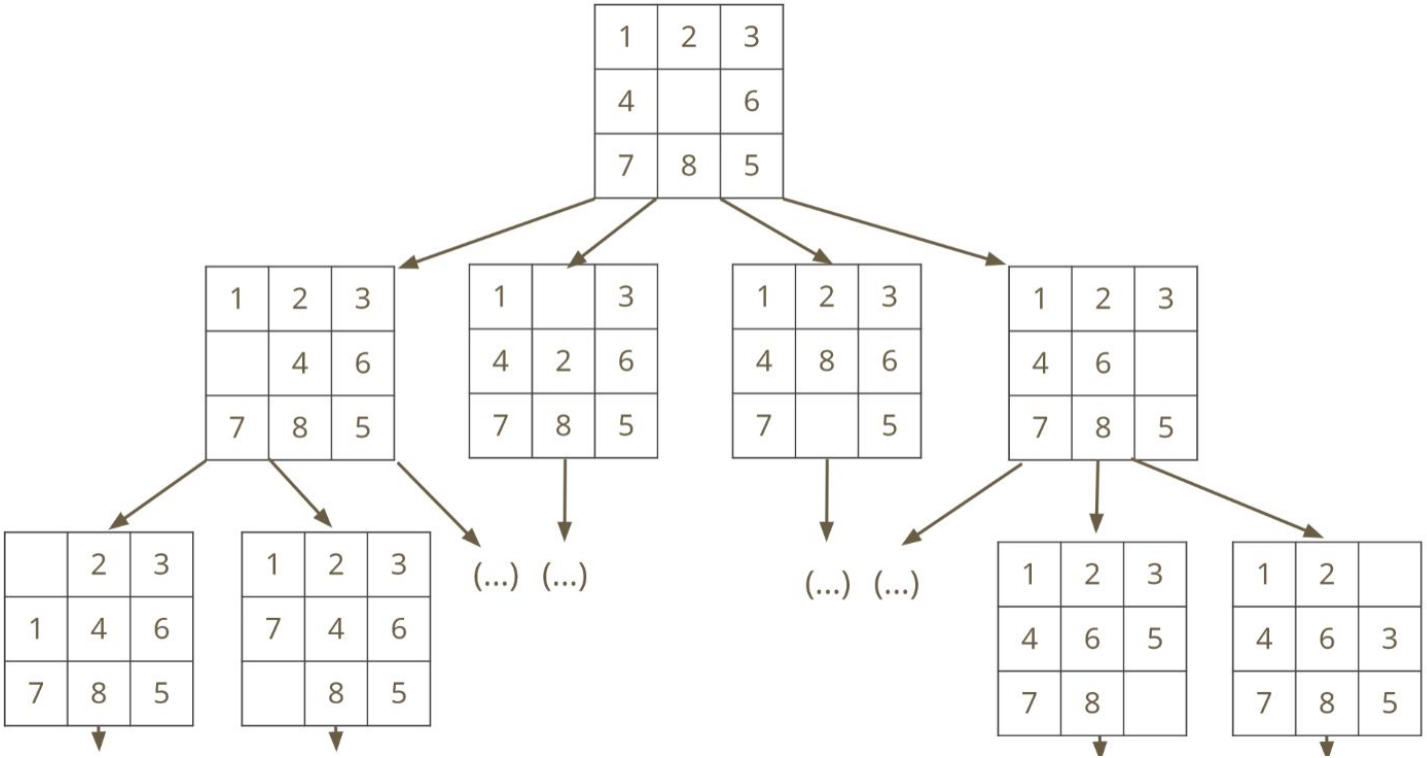
¿Cuáles son las acciones, cuáles los estados?



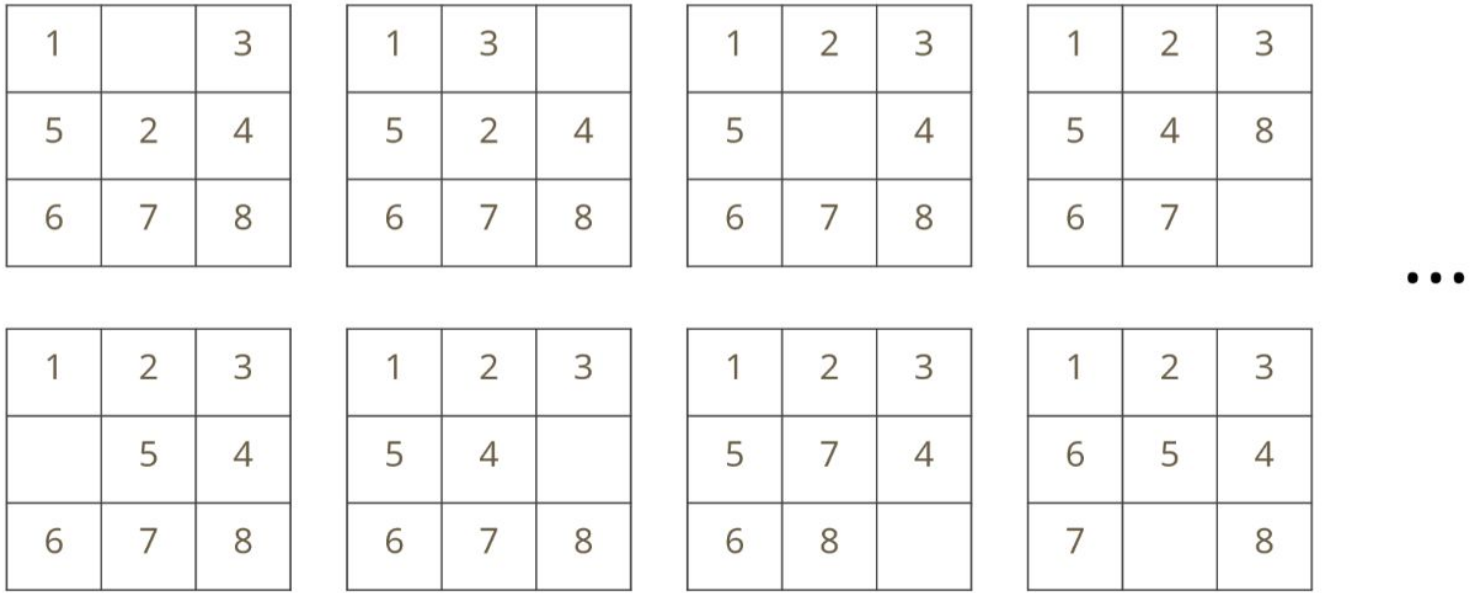
¿Cuáles son las acciones, cuáles los estados?



Grafo de Búsqueda:



Espacio de Búsqueda:



Pregunta: ¿Cuál es la cardinalidad del espacio de búsqueda?

¿Cómo resolver un problema de búsqueda?

Algoritmos de búsqueda

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

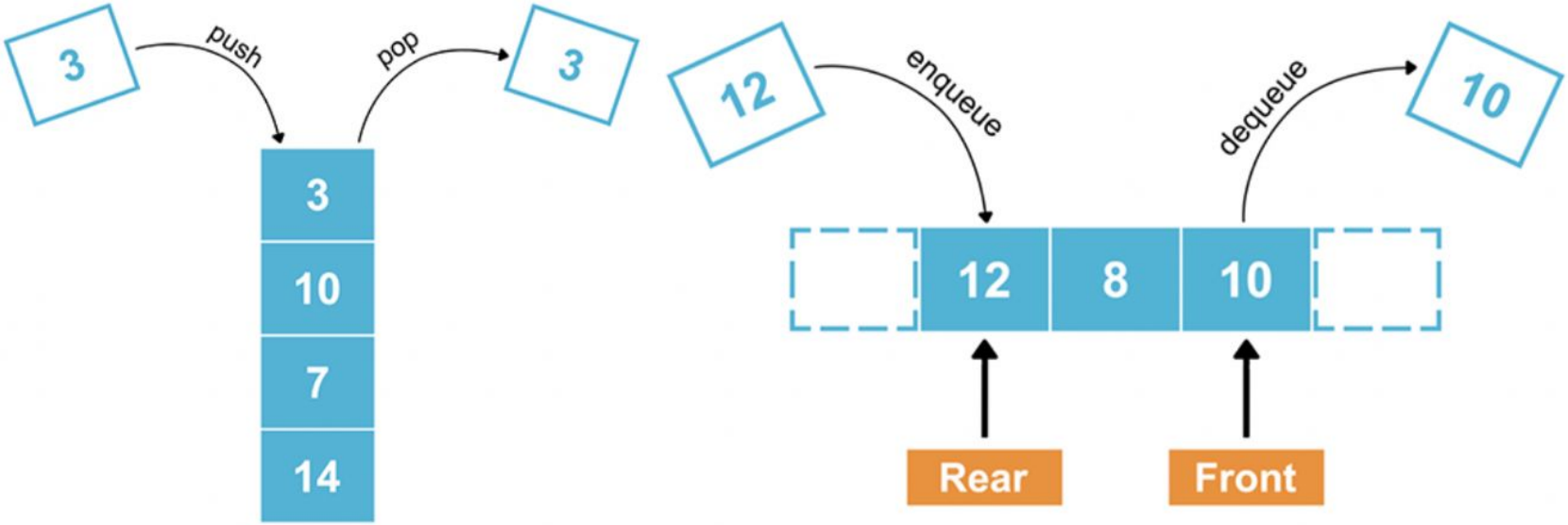
¿Cuál es la diferencia entre BFS y DFS?

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while *Open* != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

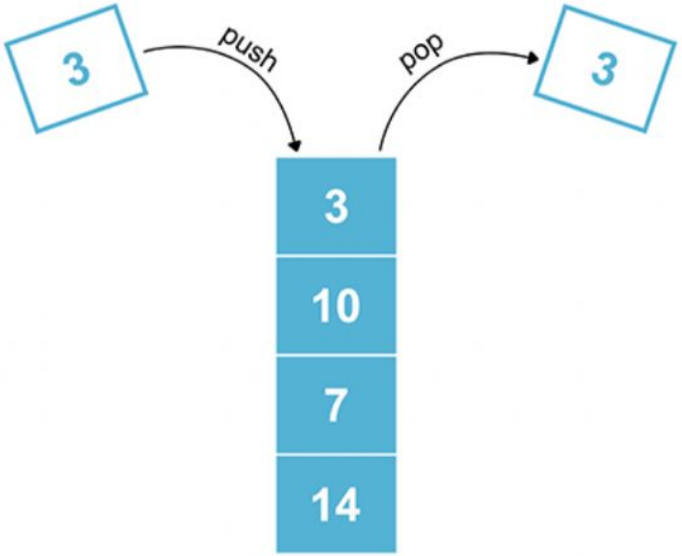
La estructura de datos que se utiliza para mantener Open

¿Cuál es la diferencia entre BFS y DFS?

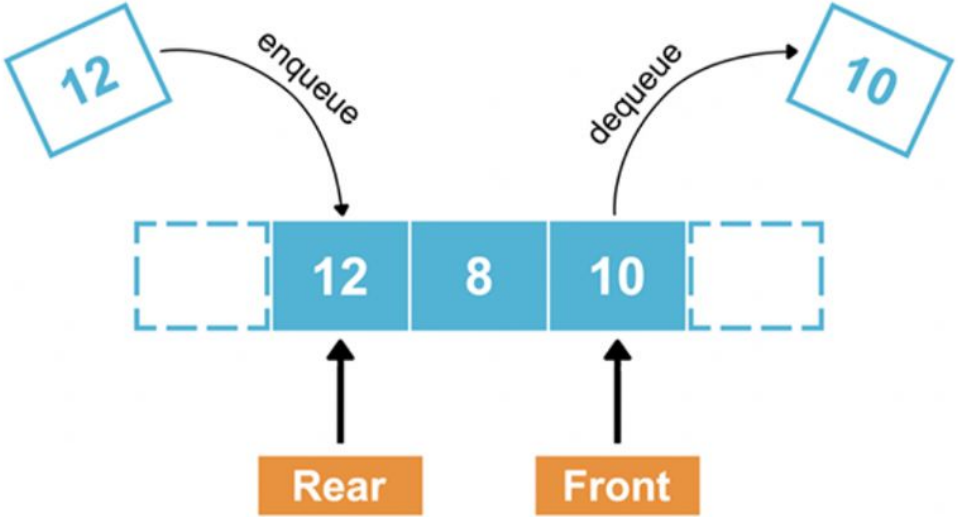


Stack **Queue**

¿Cuál es la diferencia entre BFS y DFS?



Stack
en DFS



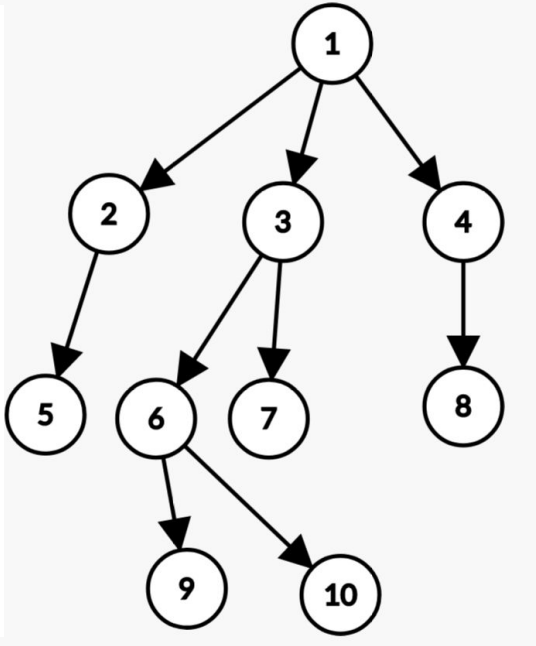
Queue
en BFS

BFS

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

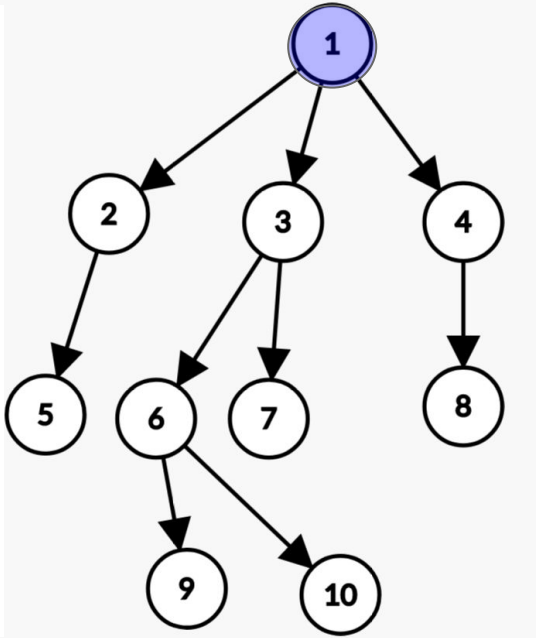



```
Closed:{}
Open:{}
Goal:{10}
```

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

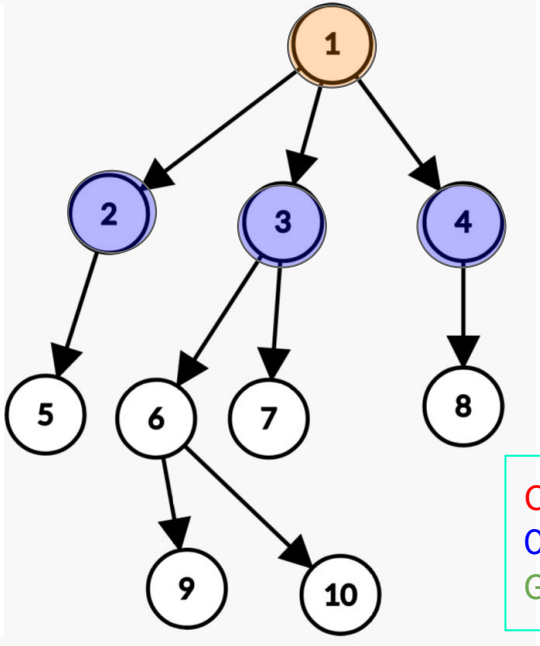


Closed:{}
Open:{1}
Goal:{10}

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

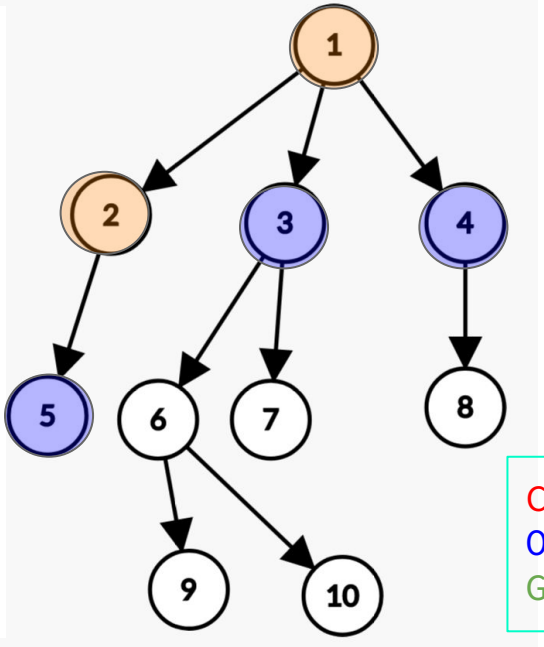


Closed:{1}
Open:{2,3,4}
Goal:{10}

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

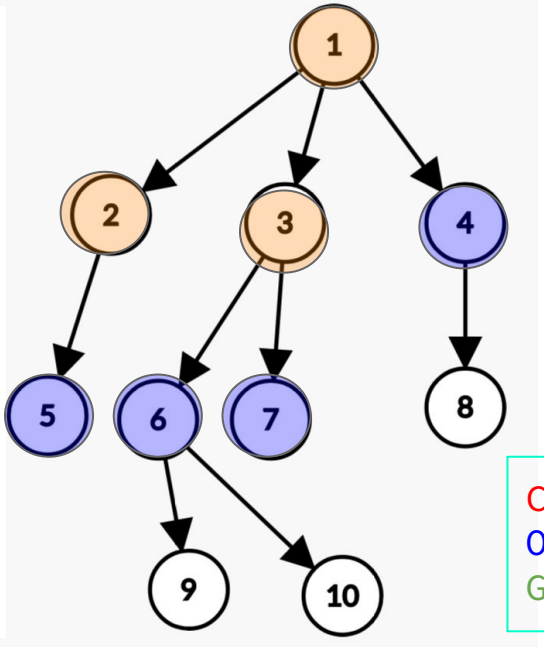


Closed:{1,2}
Open:{3,4,5}
Goal:{10}

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

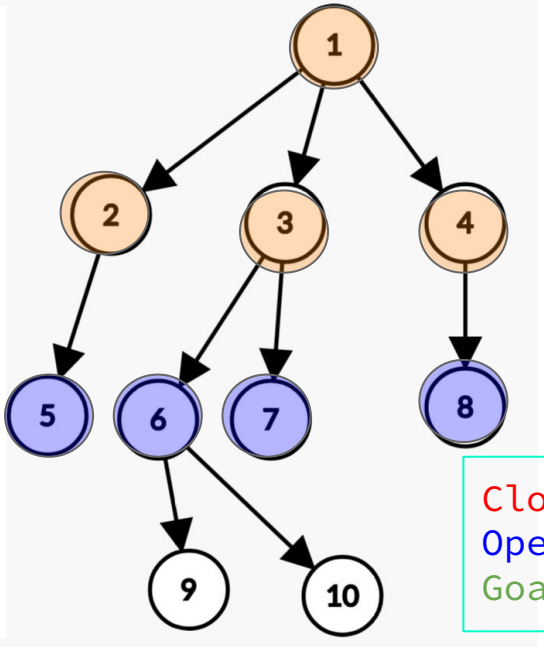


Closed:{1,2,3}
Open:{4,5,6,7}
Goal:{10}

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

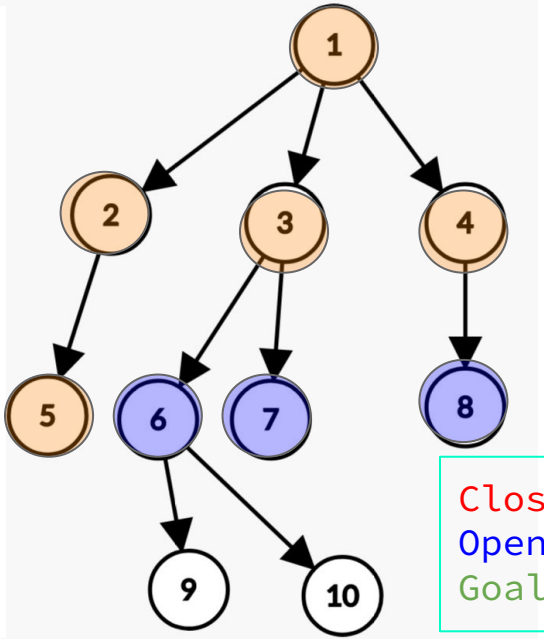


Closed:{1,2,3,4}
Open:{5,6,7,8}
Goal:{10}

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

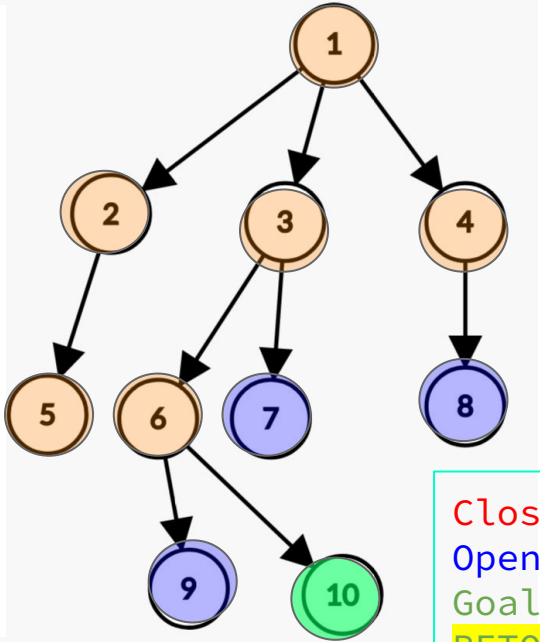


Closed:{1,2,3,4,5}
Open:{6,7,8}
Goal:{10}

BFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*



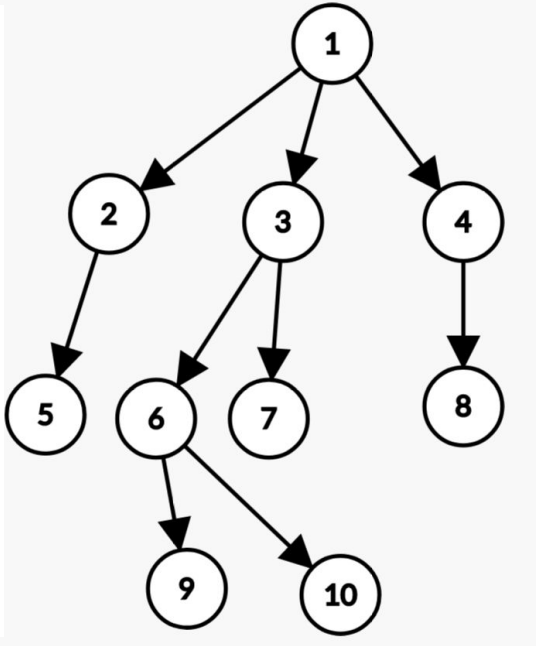

```
Closed:{1,2,3,4,5,6}
Open:{7,8,9}
Goal:{10}
RETORNA NODO 10
```

DFS

DFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

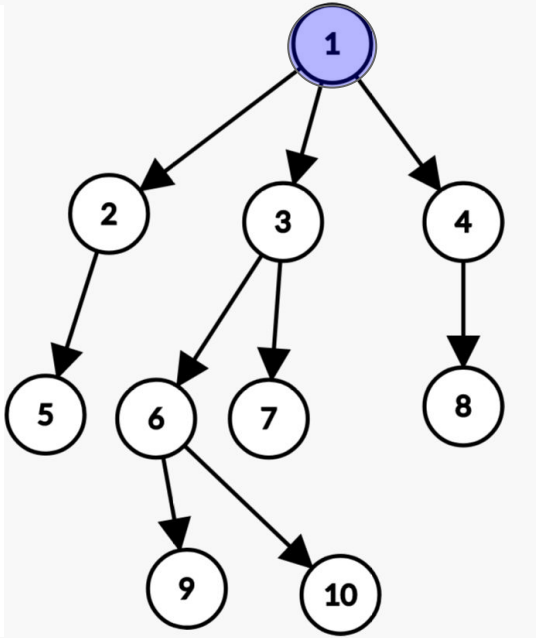



```
Closed:{}
Open:{}
Goal:{10}
```

DFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

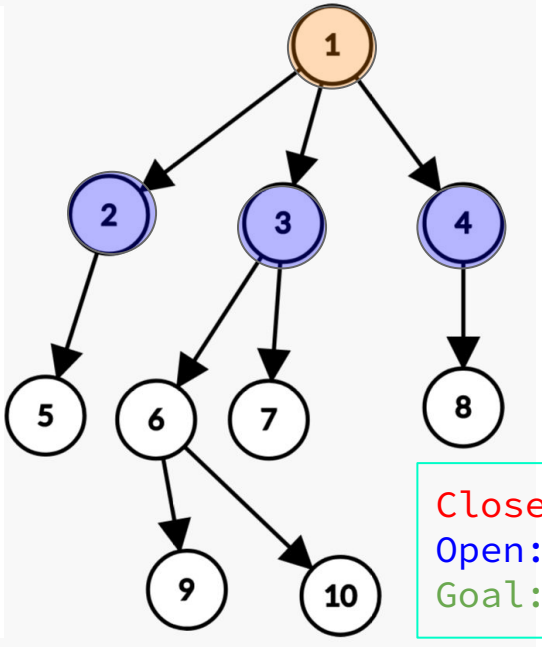


Closed:{}
Open:{1}
Goal:{10}

DFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

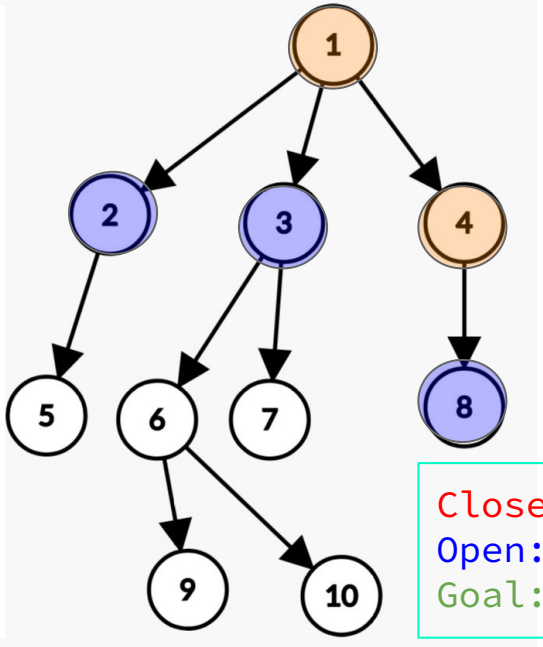


Closed:{1}
Open:{2,3,4}
Goal:{10}

DFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

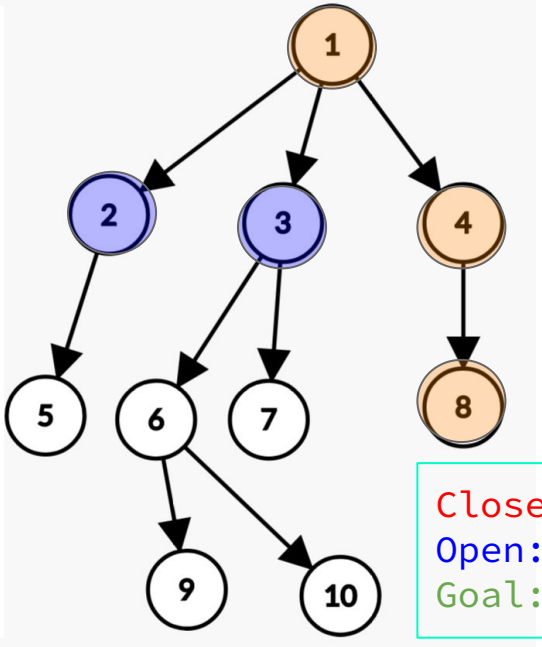


Closed:{1,4}
Open:{2,3,8}
Goal:{10}

DFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

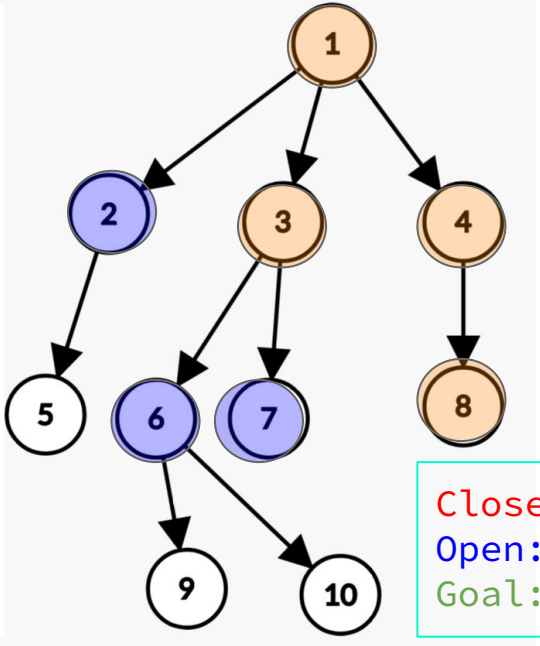


Closed:{1,4,8}
Open:{2,3}
Goal:{10}

DFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

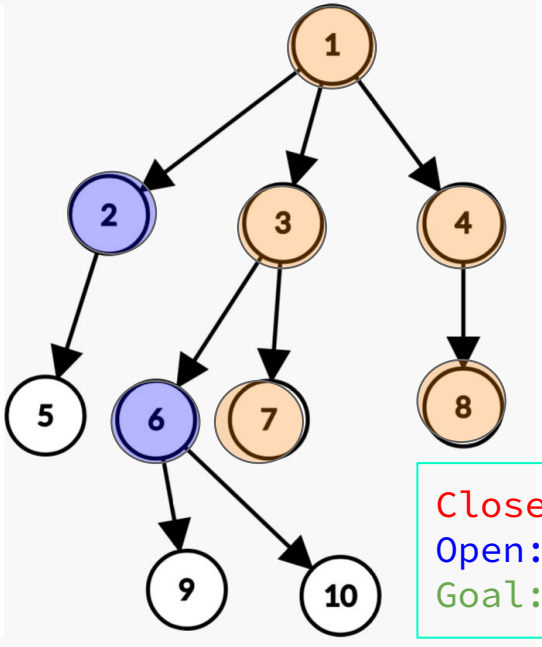


Closed:{1,4,8,3}
Open:{2,6,7}
Goal:{10}

DFS

El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

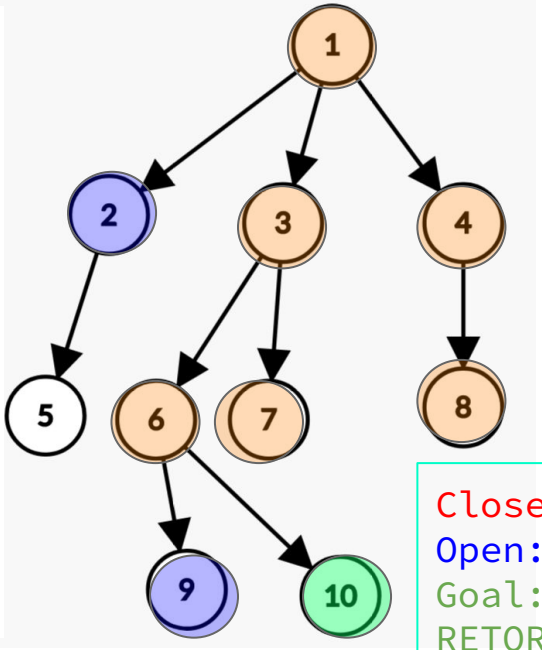


Closed:{1,4,8,3,7}
Open:{2,6}
Goal:{10}

DFS

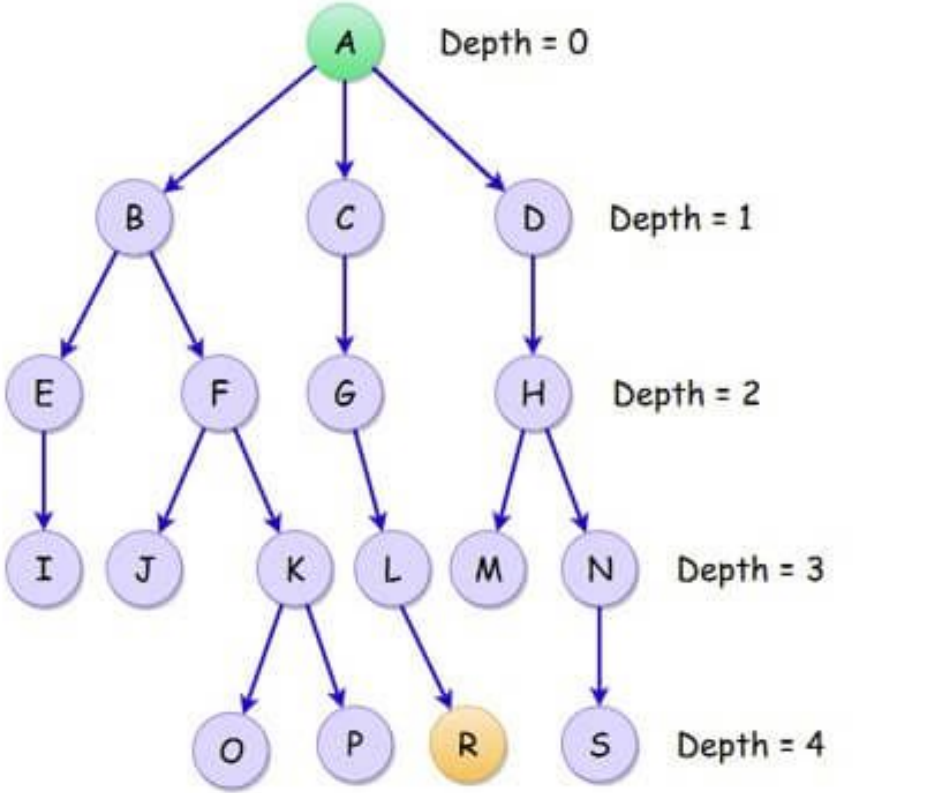
El siguiente es un algoritmo de búsqueda genérico.
Input: Un problema de búsqueda (S, A ,s_{init}, G)
Output: Un nodo objetivo

Open es un contenedor vacío
Closed es un conjunto vacío
Inserta s_{init} a *Open*
parent(s_{init}) = null
while Open != ∅:
    u ← Extraer(*Open*)
    Inserta *u* en *Closed*
    for each *v* ∈ Succ(*u*) \ (*Open* ∪ *Closed*)
        parent(*v*) = *u*
        if *v* ∈ *G* **return** *v*
        Inserta *v* a *Open*

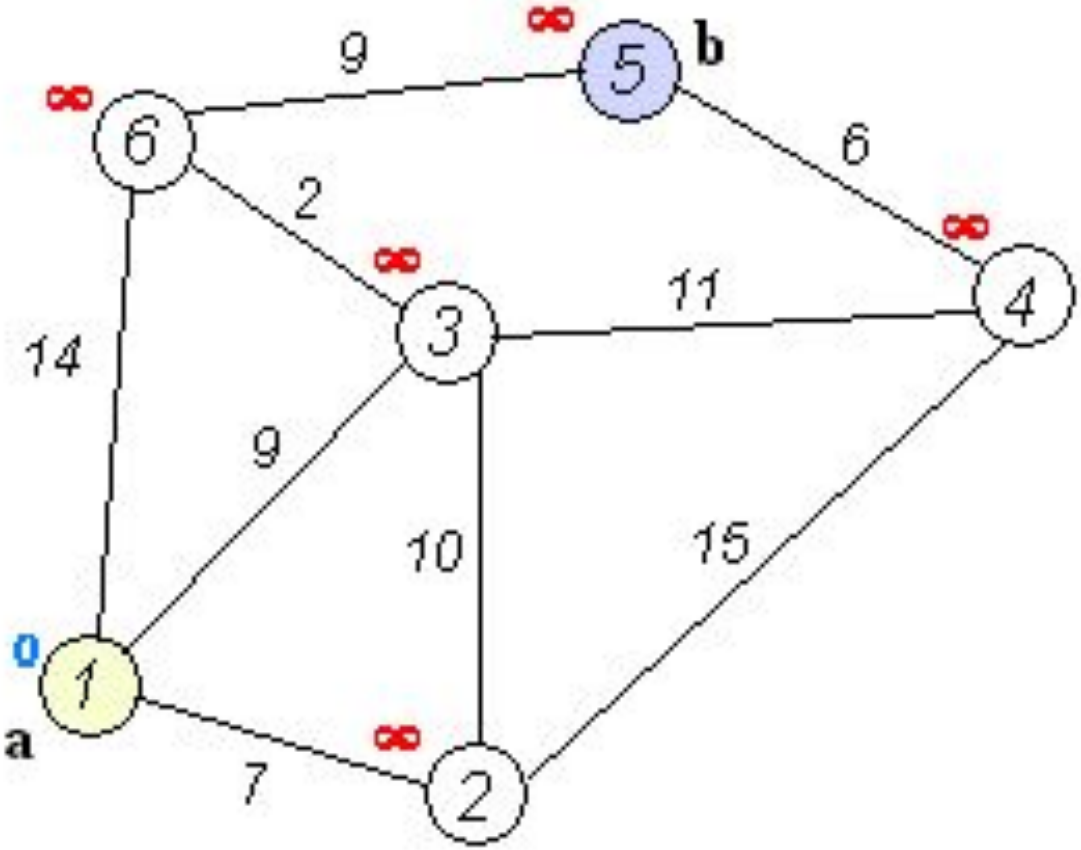


Closed:{1,4,8,3,7,6}
Open:{2,9}
Goal:{10}
RETORNAMOS NODO 10

IDDFS


A
Depth = 0
B
C
D
Depth = 1
E
F
G
H
Depth = 2
I
J
K
L
M
N
Depth = 3
O
P
R
S
Depth = 4
www.educba.com

DIJKSTRA


∞
9
5
b
∞
6
6
2
∞
∞
11
4
3
14
9
10
15
0
1
a
∞
7
2

¡Vayamos al notebook!

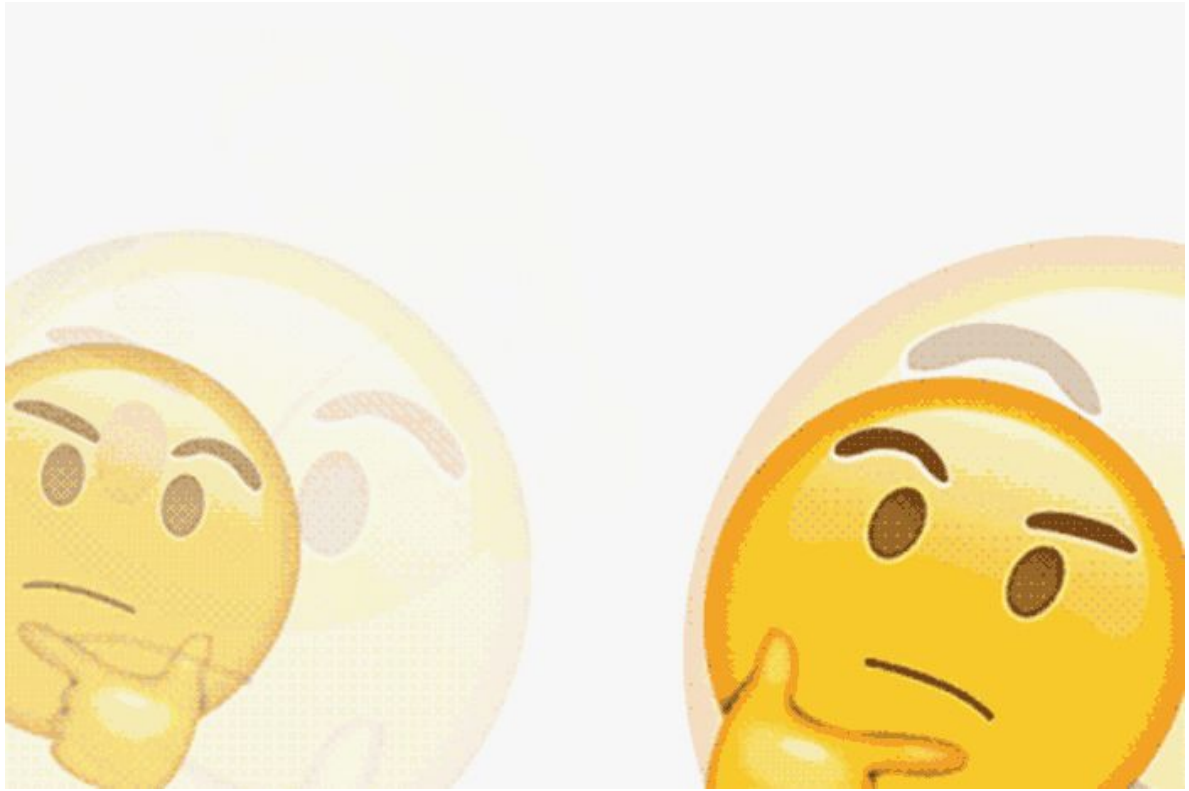
Otro Ejemplo...

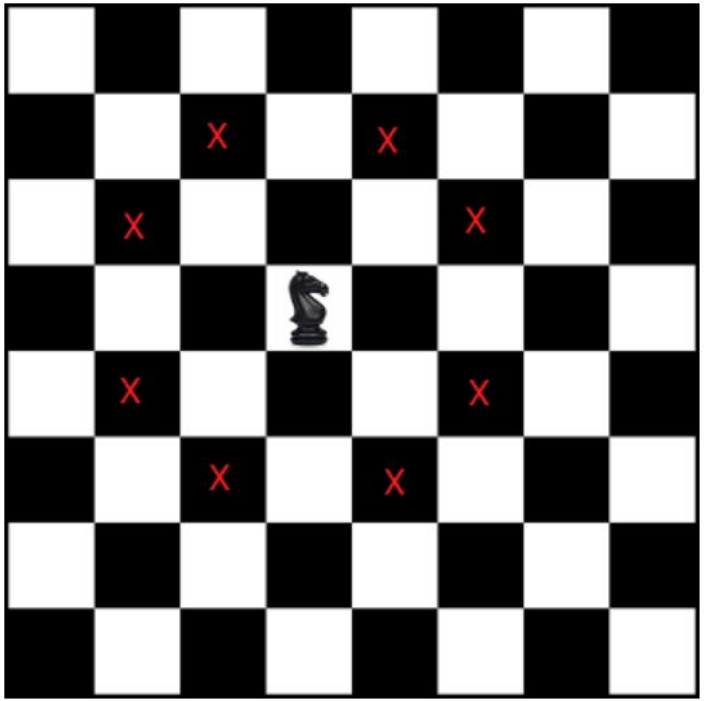
Un ejemplo:

Nos vamos a centrar en una sola pieza:

Ejemplo de u

¿Cómo se mueve?





Buscar un camino para el caballo, dados dos cuadros del tablero