

---

---

# Ayudantía 4 - Búsqueda

— IIC2613 - Inteligencia Artificial —

---

---

# Búsqueda



# Terminología

- Estado: configuración específica del sistema.
- Acción -  $a$ : una acción que transforma el estado actual del sistema.
- Conjunto de Acciones -  $\mathcal{A}$ : Todas las acciones posibles.
- Espacio de búsqueda -  $\mathcal{S}$ : Conjunto de todos los estados posibles
- Grafo de búsqueda: Todos los estados posibles conectados por las acciones que los unen.
- Problema de búsqueda ( $\mathcal{S}, \mathcal{A}, s_{init}, G$ )

# Puzzle 8



# Puzzle 8

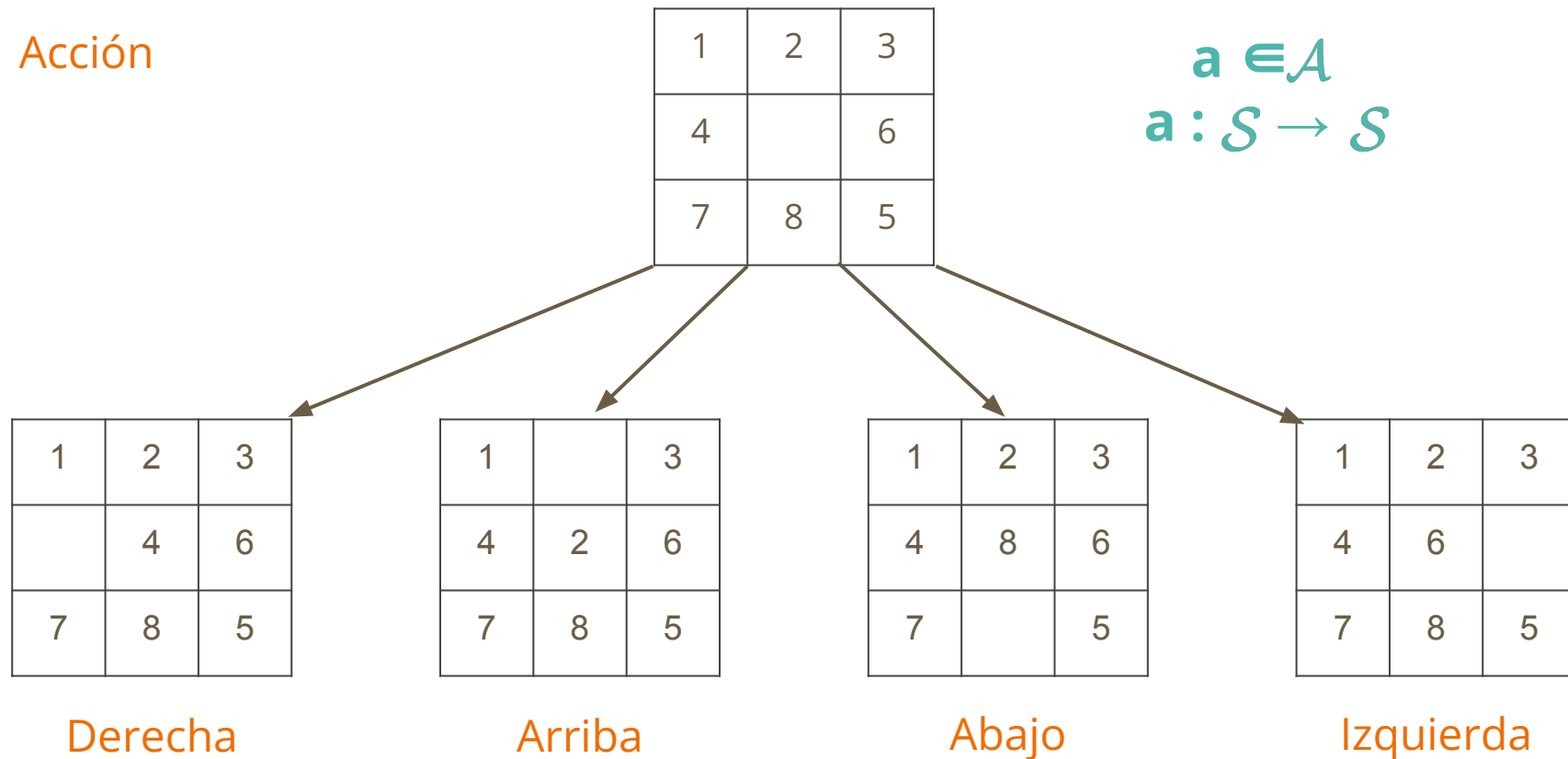
Estado

1	2	3
5		4
6	7	8

$s \in S$

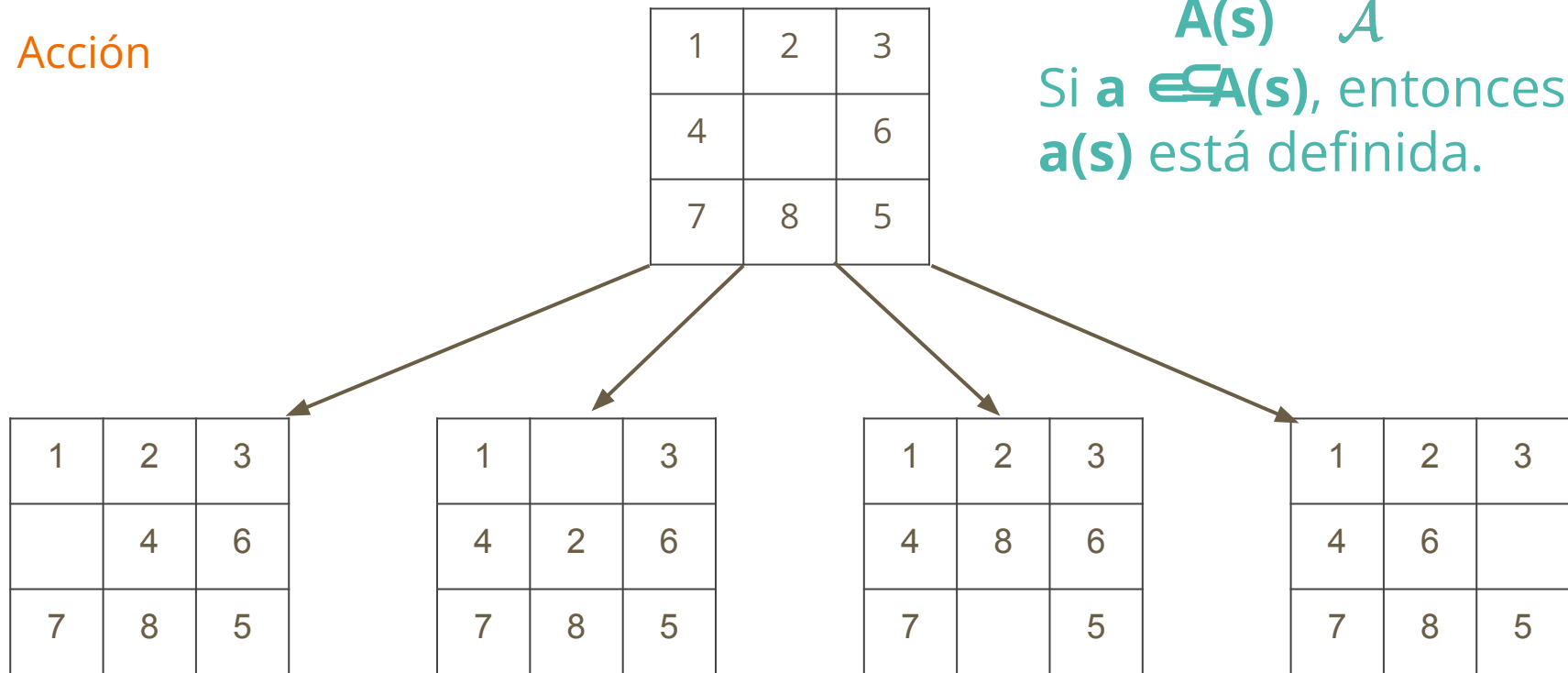
# Puzzle 8

Acción



# Puzzle 8

Acción



# Puzzle 8

Acción

1	2	3
4		6
7	8	5

$$\text{Succ}(s) = \{ a(s) \mid a \in A(s) \}$$

1	2	3
	4	6
7	8	5

1		3
4	2	6
7	8	5

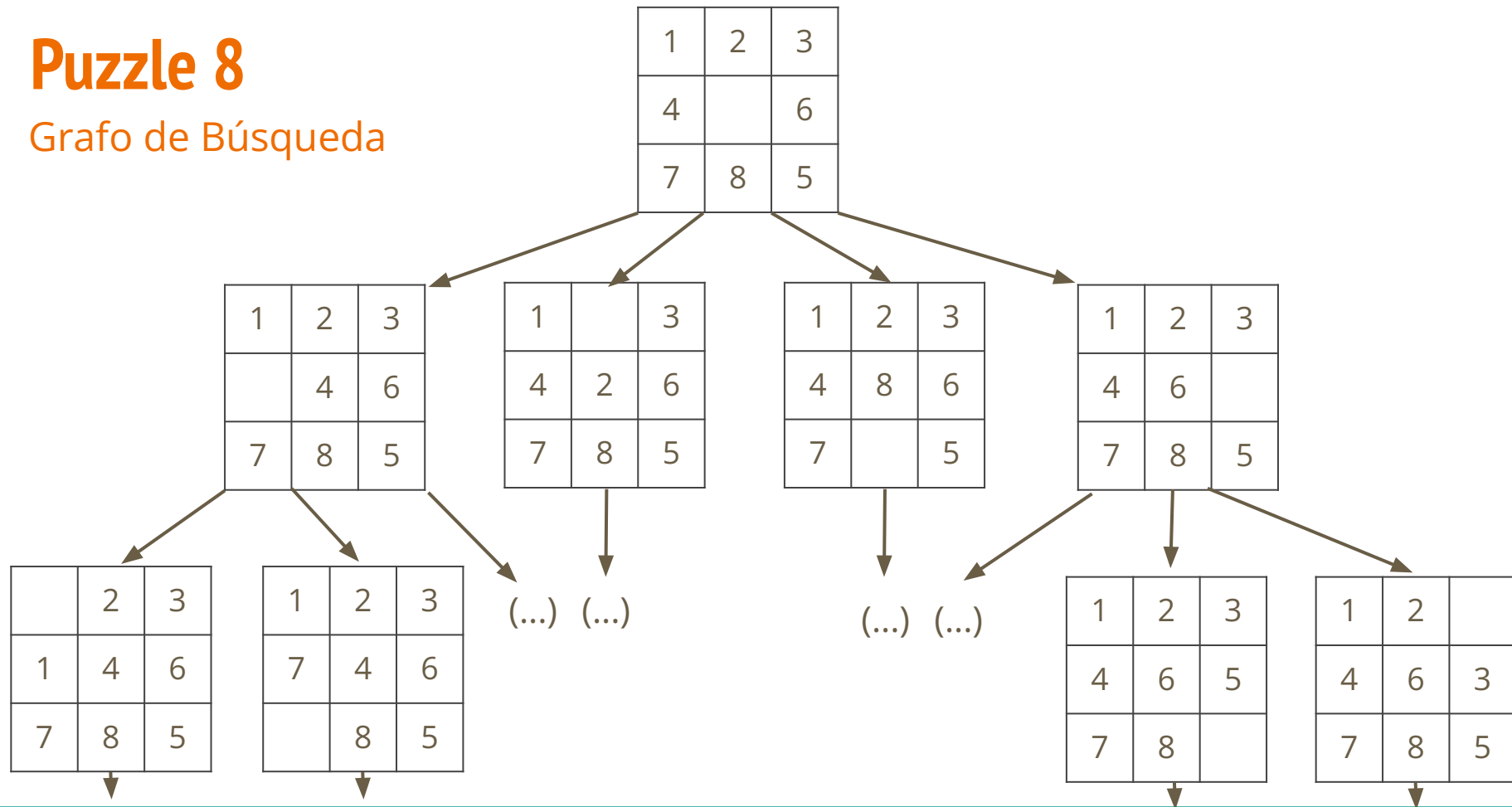
1	2	3
4	8	6
7		5

1	2	3
4	6	
7	8	5



# Puzzle 8

## Grafo de Búsqueda



# Puzzle 8

## Espacio de Búsqueda

1		3
5	2	4
6	7	8

1	3	
5	2	4
6	7	8

1	2	3
5		4
6	7	8

1	2	3
5	4	8
6	7	

...

1	2	3
	5	4
6	7	8

1	2	3
5	4	
6	7	8

1	2	3
5	7	4
6	8	

1	2	3
6	5	4
7		8

# Puzzle 8

Espacio de Búsqueda

# Cantidad de estados Puzzle 8 =  $9!$  = **362.880**

# Puzzle 8

Espacio de Búsqueda

# Cantidad de estados Puzzle 8 =  $9!$  = **362.880**

# Cantidad de estados Puzzle 16 =  $16!$  = **20.922.789.888.000**

# Puzzle 8

## Problema de búsqueda

- Problema de búsqueda ( $S, A, s_{init}, G$ )
  - $S$  = conjunto de estados
  - $A$  = 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  $\in G$

# Algoritmos de Búsqueda

- BFS
- DFS
- IDDFS
- Dijkstra

# Algoritmo de Búsqueda Genérico

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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\textit{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\textit{Open} \cup \textit{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

        Inserta  $v$  a *Open*

# Algoritmo de Búsqueda Genérico

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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\textit{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\textit{Open} \cup \textit{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

        Inserta  $v$  a *Open*



En **BFS** se usa una **Cola**



# Algoritmo de Búsqueda Genérico

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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\textit{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\textit{Open} \cup \textit{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

        Inserta  $v$  a *Open*

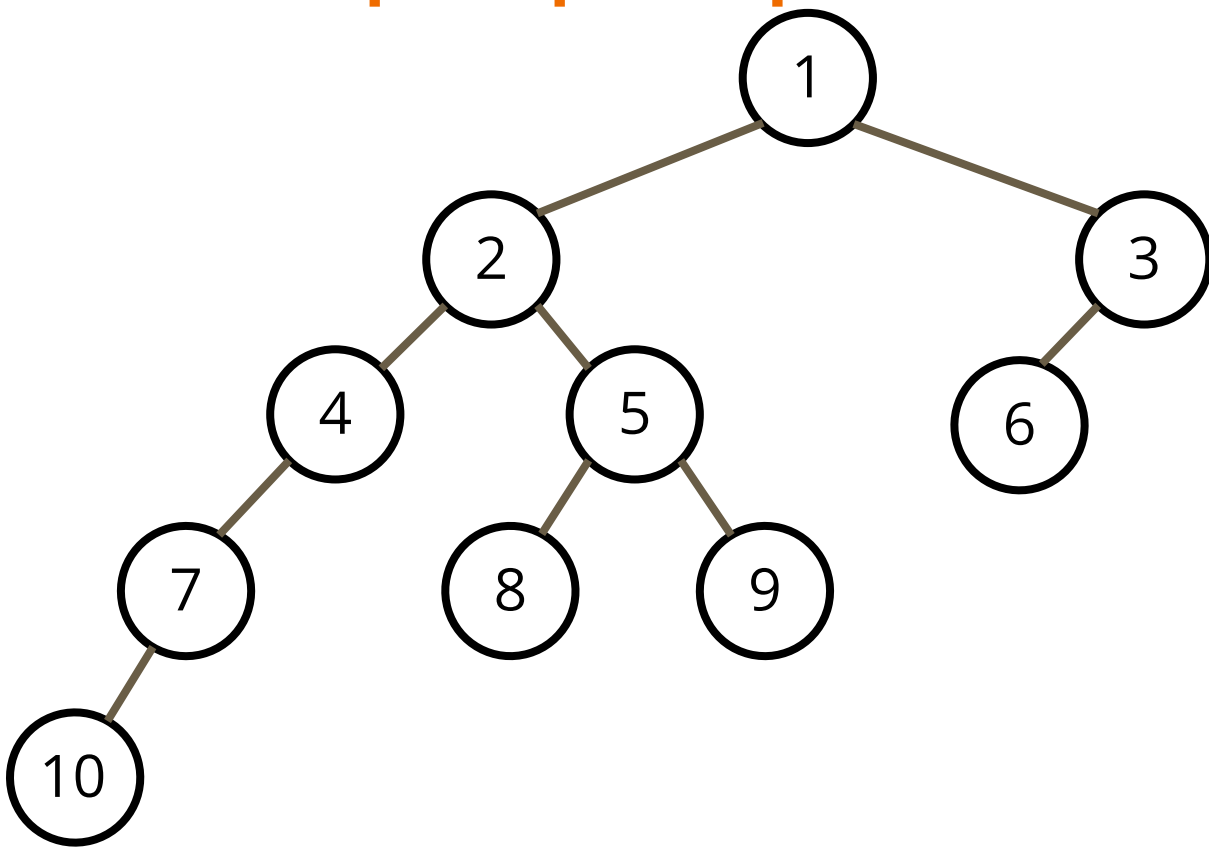


En **DFS** se usa un **Stack**

# BFS

Búsqueda por Amplitud

# BFS - Búsqueda por Amplitud



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*

$parent(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$parent(v) = u$

**if**  $v \in G$  **return**  $v$

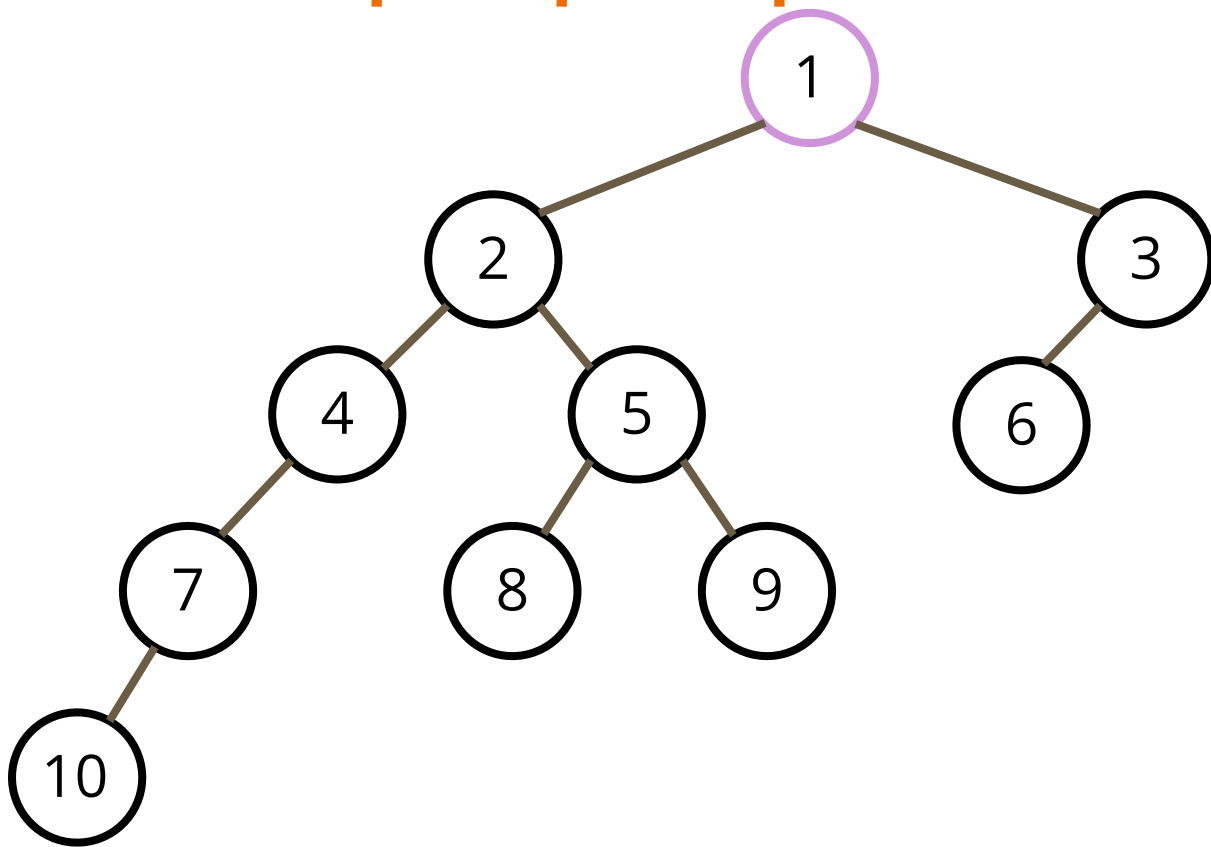
        Inserta  $v$  a *Open*

**Closed:**  $[]$

**Open:**  $[]$

**Goal:**  $\{9\}$

# BFS - Búsqueda por Amplitud



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*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$parent(v) = u$

**if**  $v \in G$  **return**  $v$

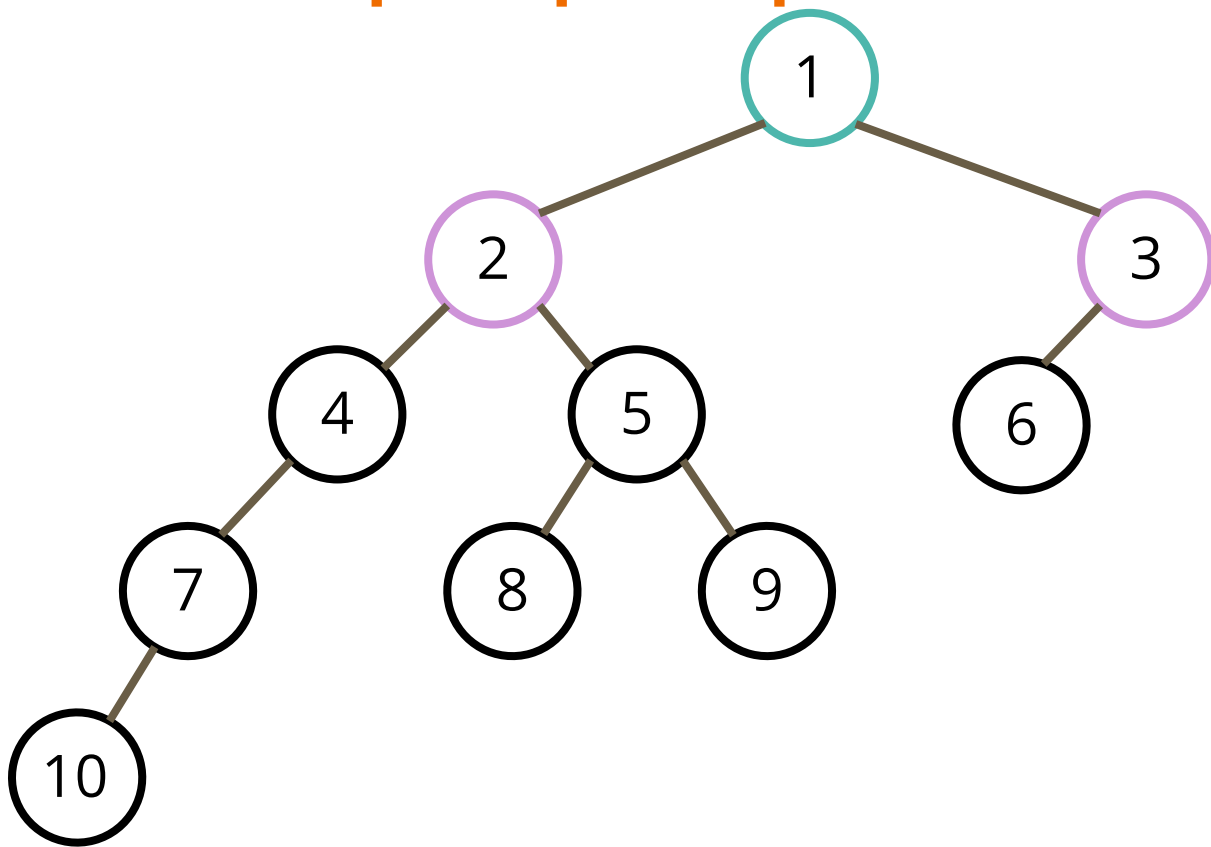
        Inserta  $v$  a *Open*

**Closed:**  $[\ ]$

**Open:**  $[1]$

**Goal:**  $\{9\}$

# BFS - Búsqueda por Amplitud



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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

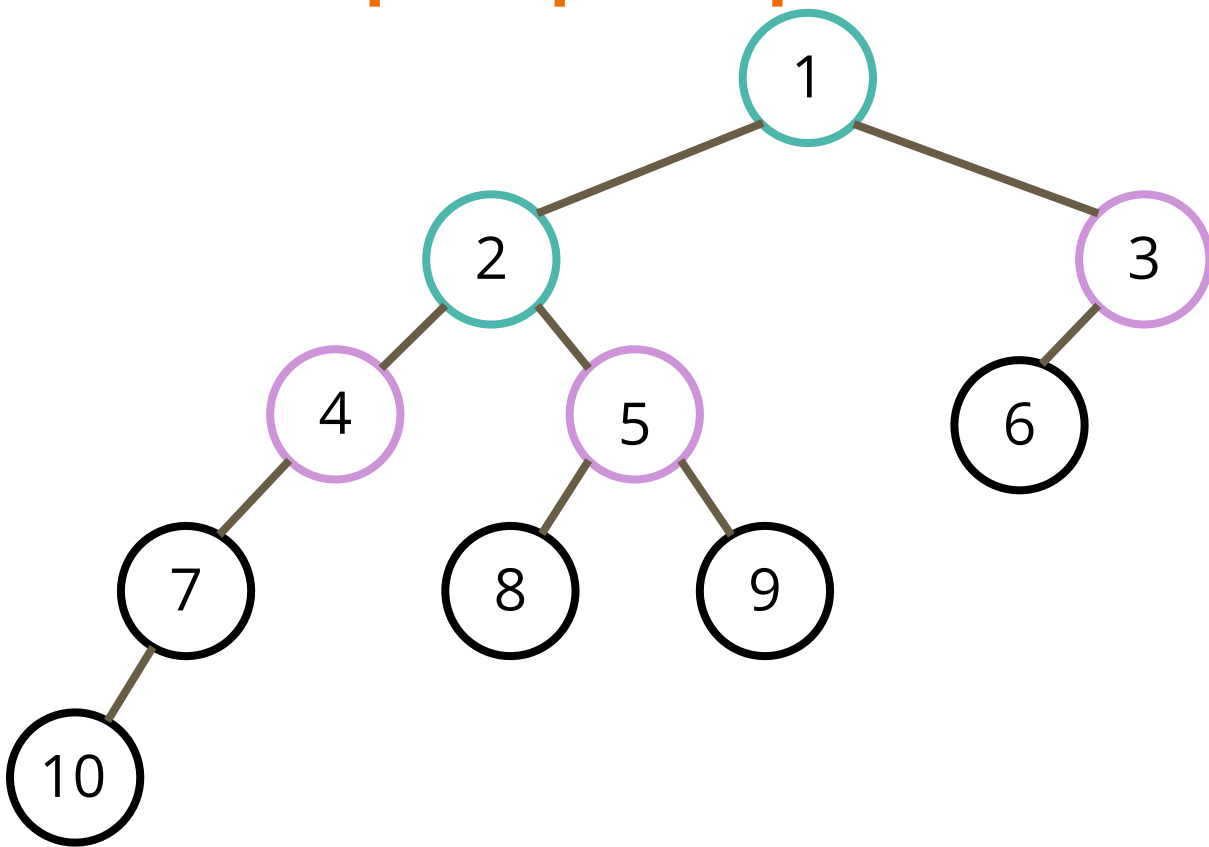
        Inserta  $v$  a *Open*

**Closed:** [1]

**Open:** [2, 3]

**Goal:** {9}

# BFS - Búsqueda por Amplitud



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*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$parent(v) = u$

**if**  $v \in G$  **return**  $v$

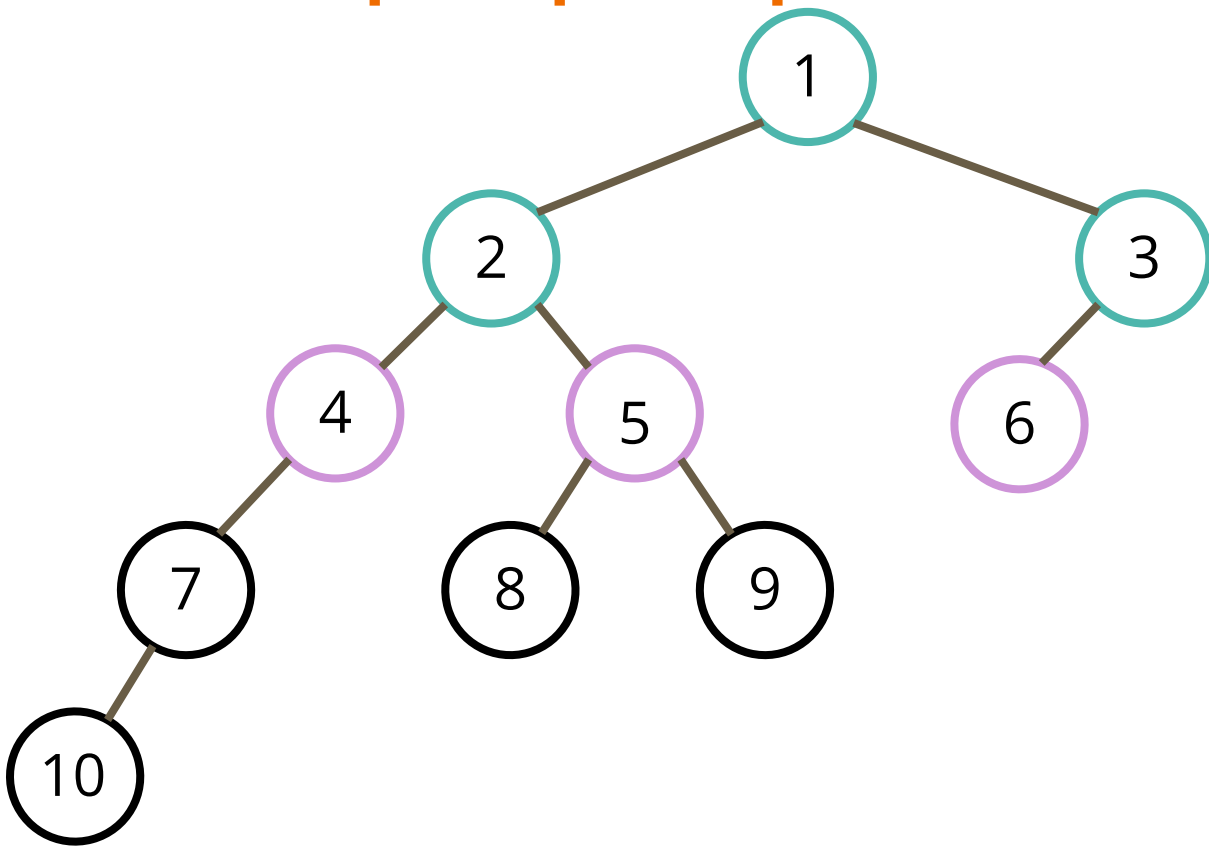
        Inserta  $v$  a *Open*

**Closed:** [1, 2]

**Open:** [3, 4, 5]

**Goal:** {9}

# BFS - Búsqueda por Amplitud



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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

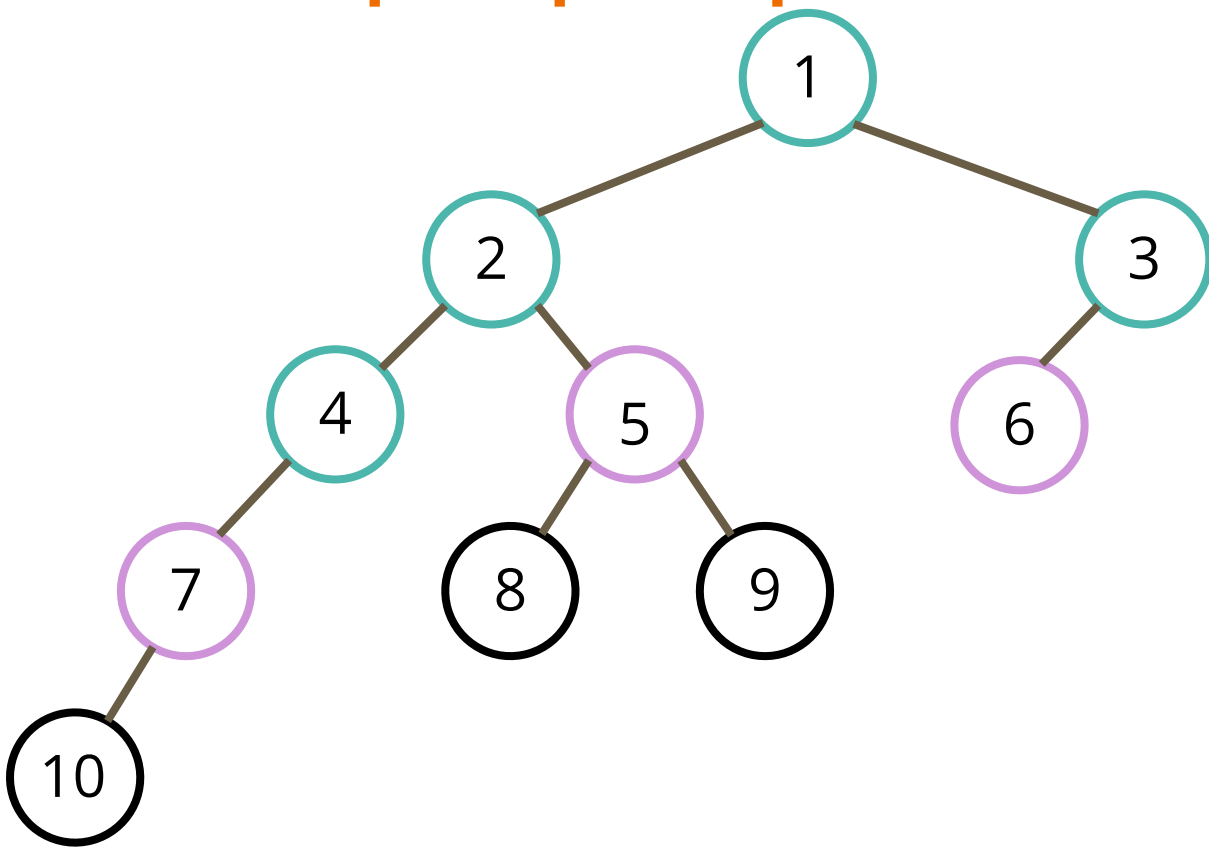
        Inserta  $v$  a *Open*

**Closed:** [1, 2, 3]

**Open:** [4, 5, 6]

**Goal:** {9}

# BFS - Búsqueda por Amplitud



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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

        Inserta  $v$  a *Open*

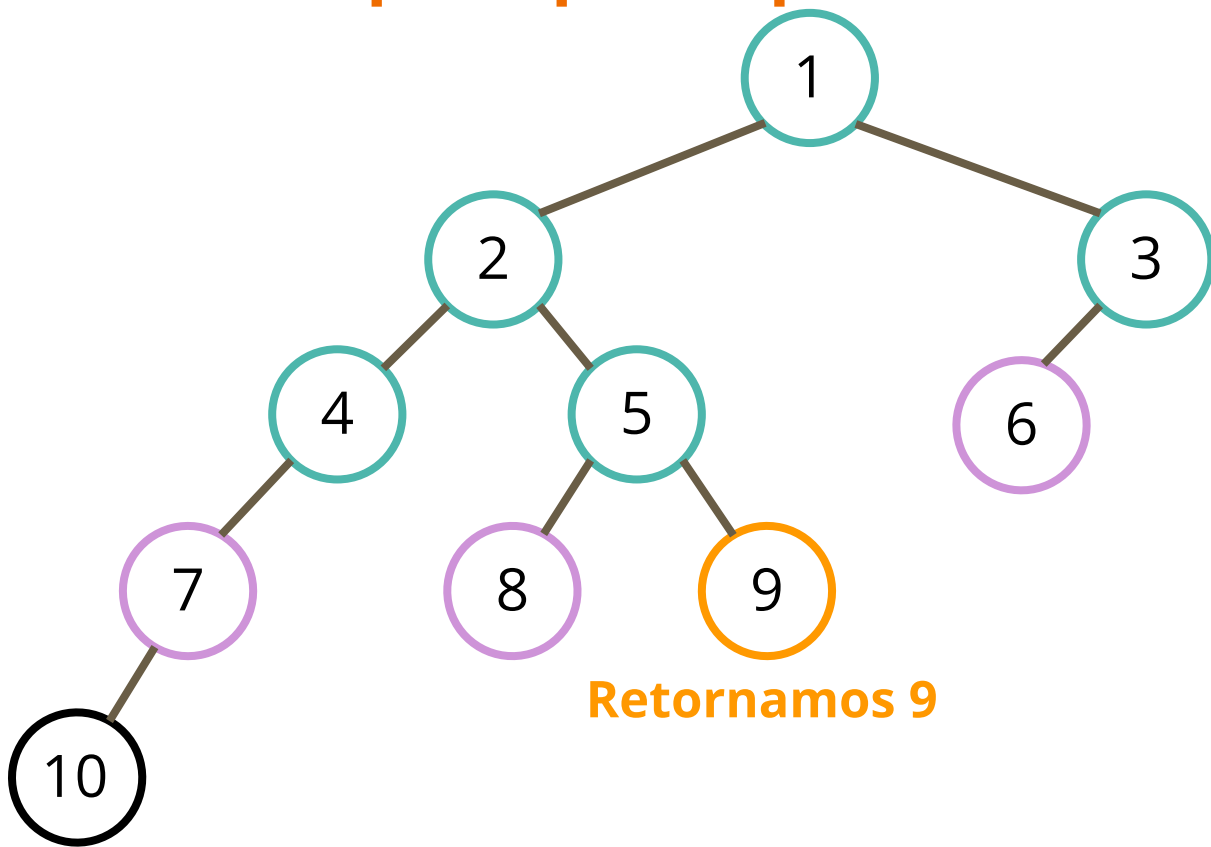
**Closed:** [1, 2, 3, 4]

**Open:** [5, 6, 7]

**Goal:** {9}



# BFS - Búsqueda por Amplitud



Retornamos 9

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*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$parent(v) = u$

**if**  $v \in G$  **return**  $v$

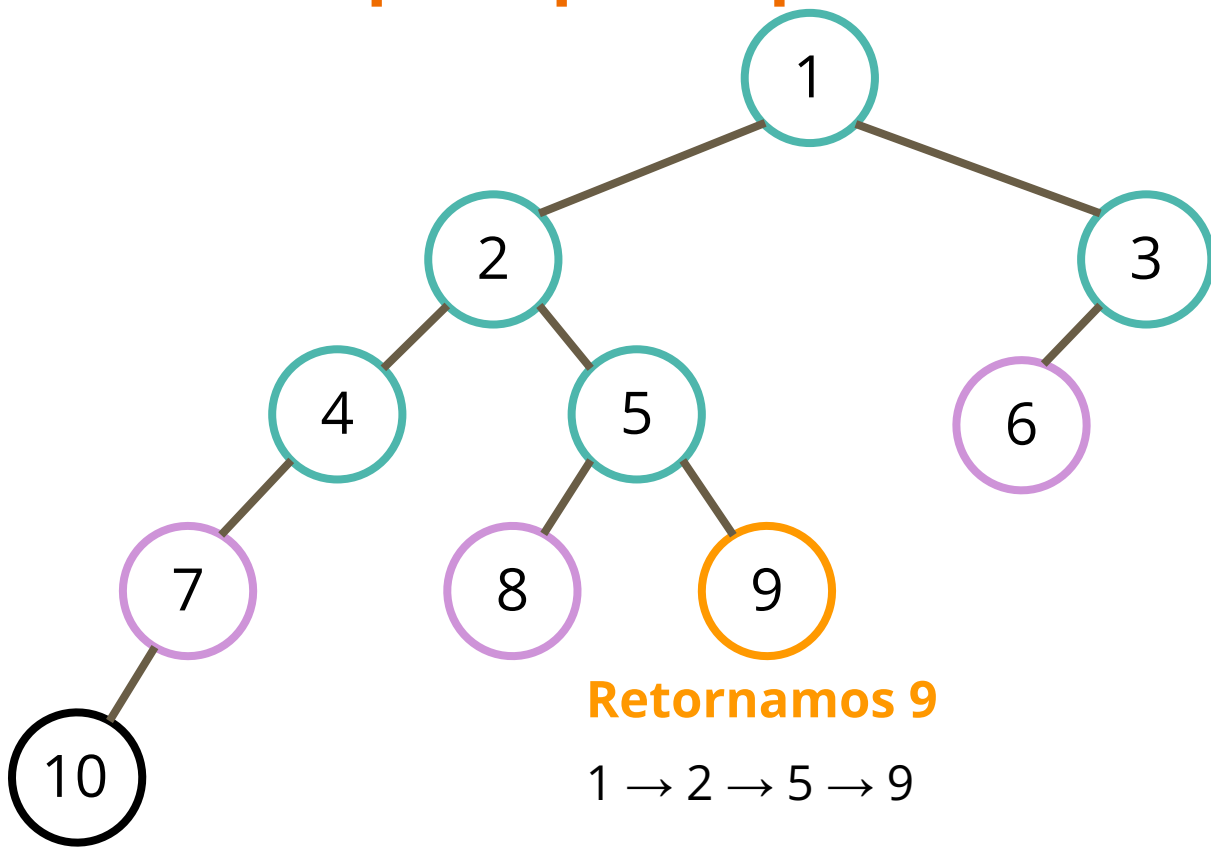
        Inserta  $v$  a *Open*

**Closed:** [1, 2, 3, 4, 5]

**Open:** [6, 7, 8]

**Goal:** {9}

# BFS - Búsqueda por Amplitud



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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

        Inserta  $v$  a *Open*

**Closed:** [1, 2, 3, 4, 5]

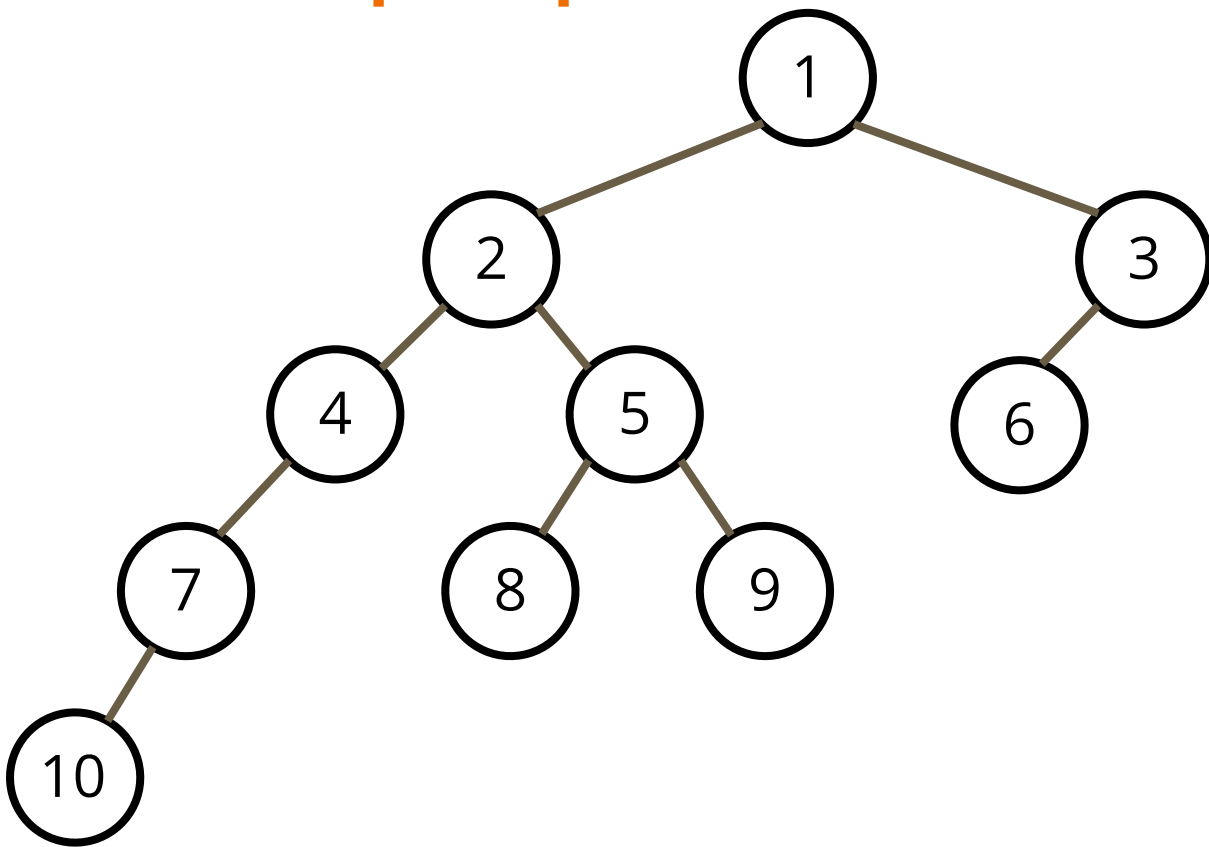
**Open:** [6, 7, 8]

**Goal:** {9}

# DFS

Búsqueda por Profundidad

# DFS - Búsqueda por Profundidad



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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

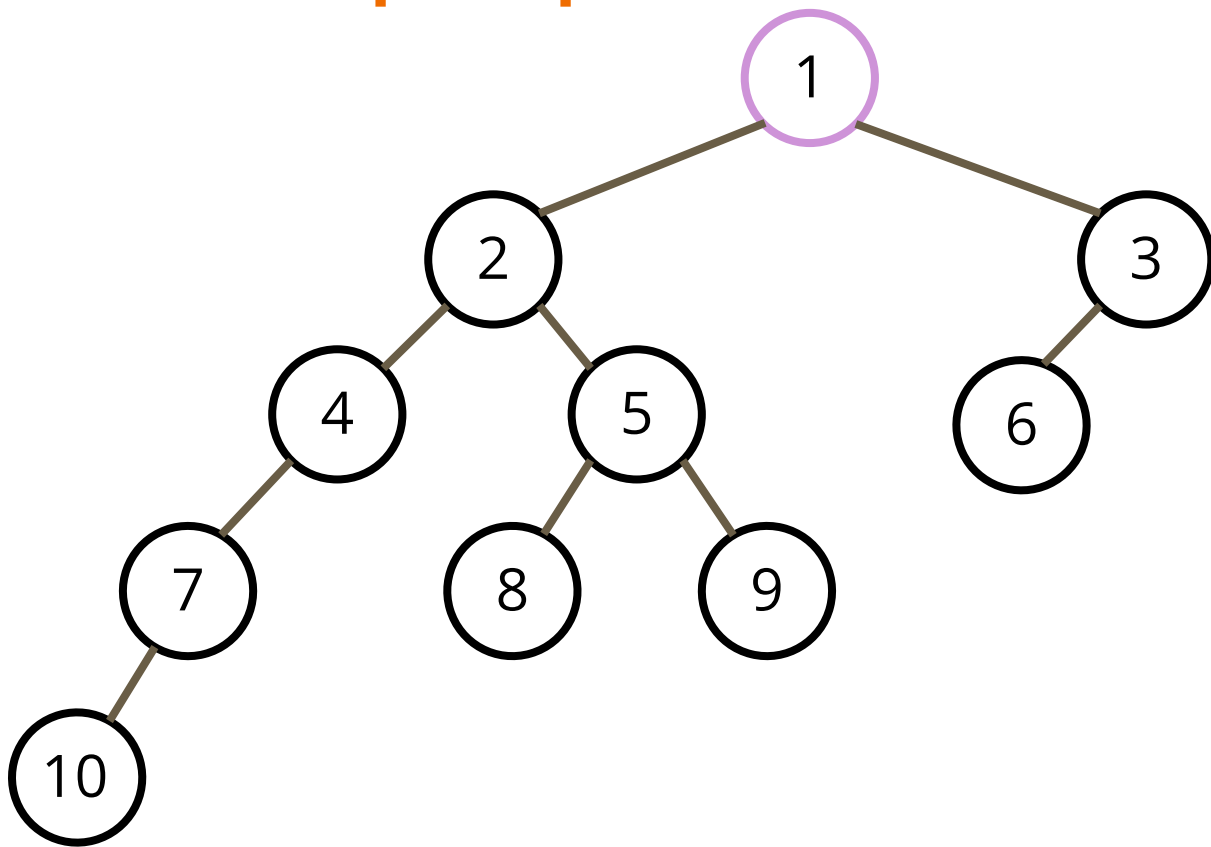
        Inserta  $v$  a *Open*

**Closed:**  $[\ ]$

**Open:**  $[\ ]$

**Goal:**  $\{9\}$

# DFS - Búsqueda por Profundidad



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*

$parent(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$parent(v) = u$

**if**  $v \in G$  **return**  $v$

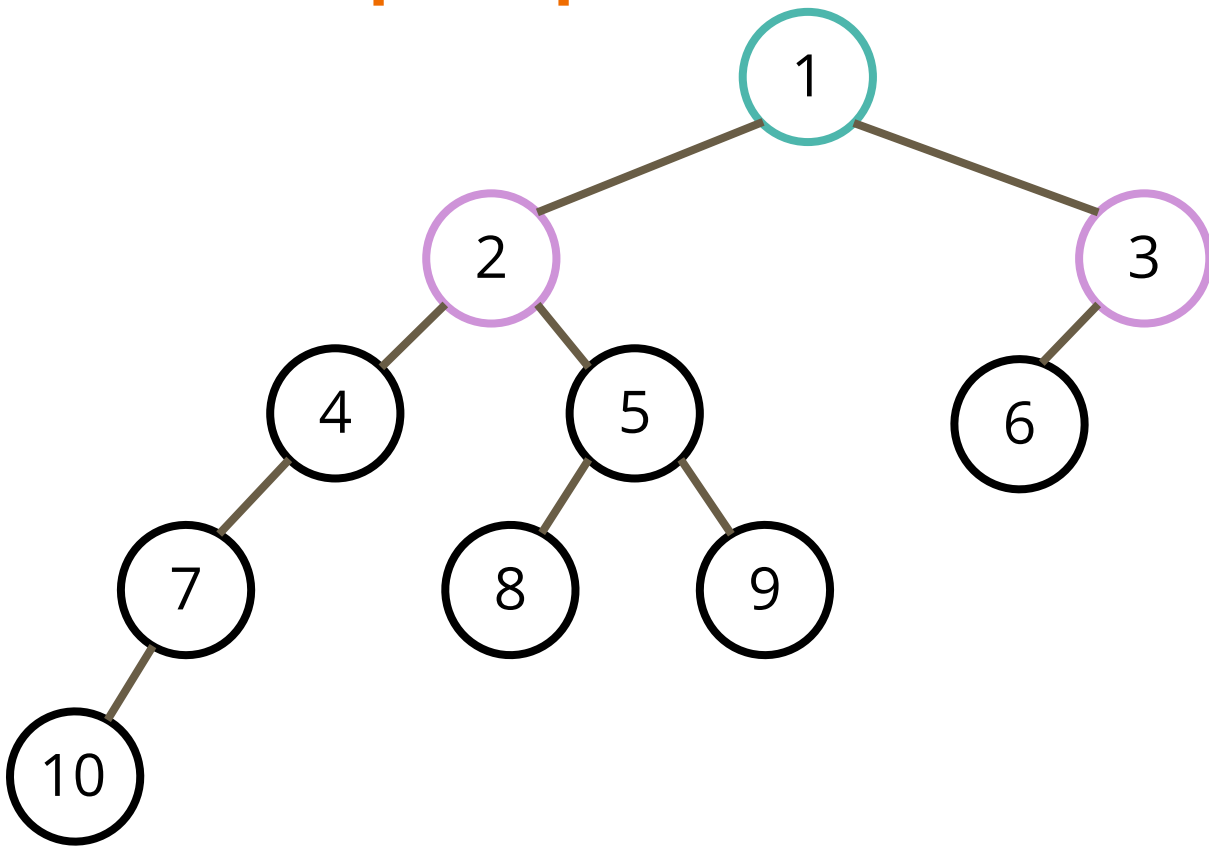
        Inserta  $v$  a *Open*

**Closed:**  $[\ ]$

**Open:**  $[1]$

**Goal:**  $\{9\}$

# DFS - Búsqueda por Profundidad



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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

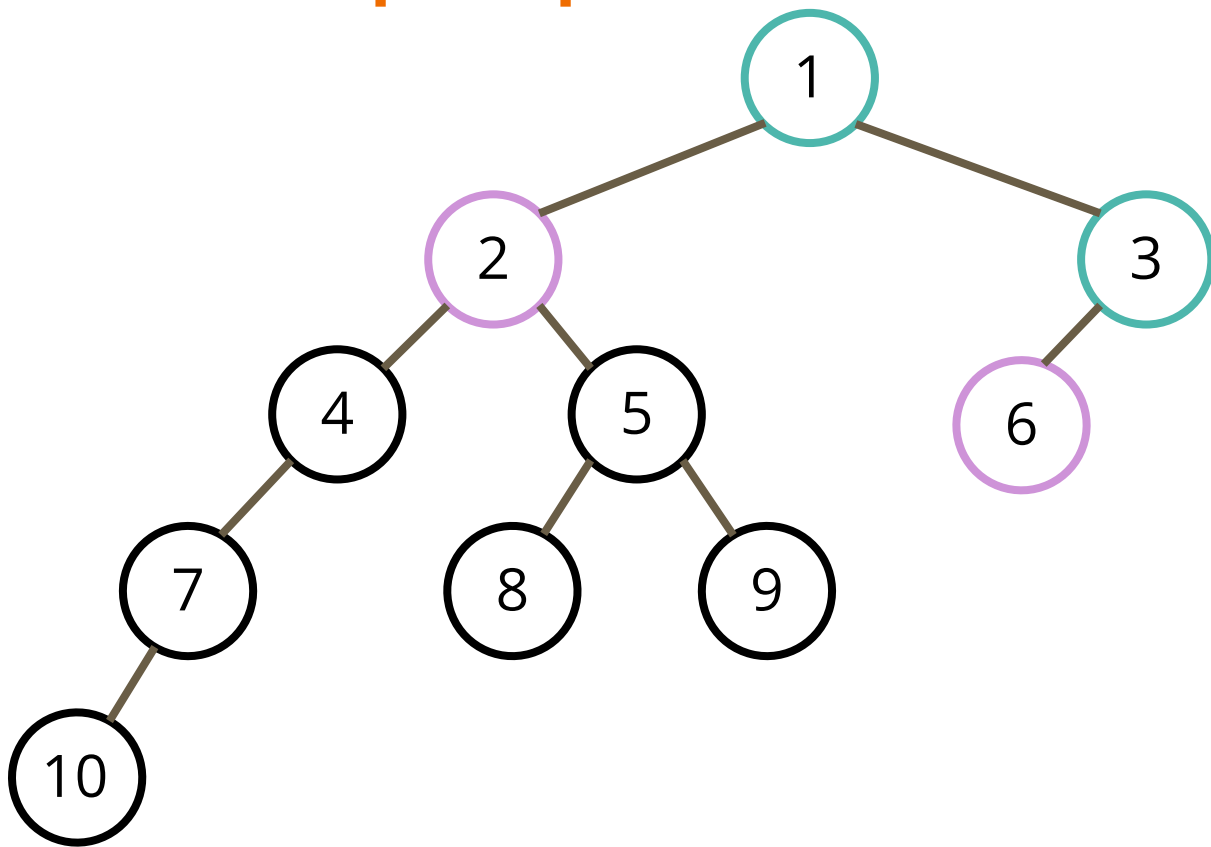
        Inserta  $v$  a *Open*

**Closed:** [1]

**Open:** [2, 3]

**Goal:** {9}

# DFS - Búsqueda por Profundidad



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*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$parent(v) = u$

**if**  $v \in G$  **return**  $v$

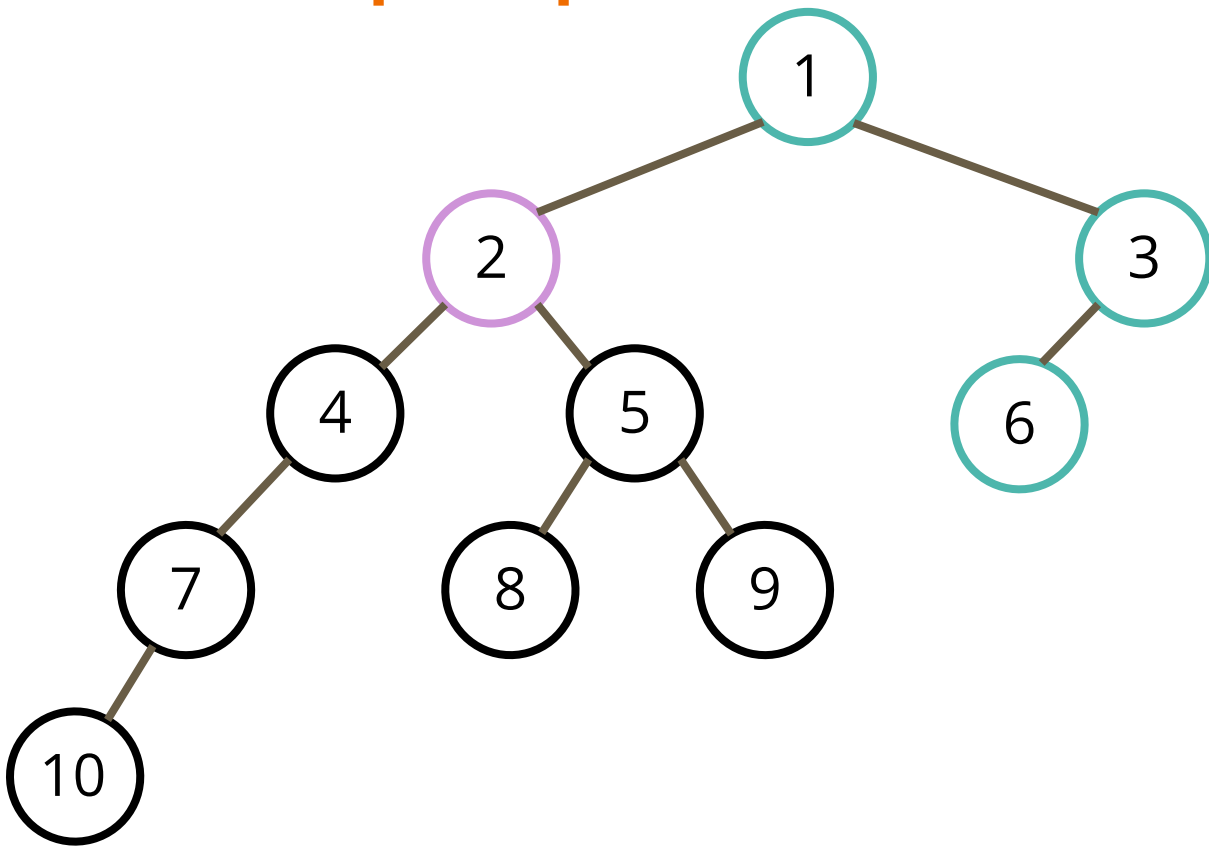
        Inserta  $v$  a *Open*

**Closed:** [1, 3]

**Open:** [2, 6]

**Goal:** {9}

# DFS - Búsqueda por Profundidad



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*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$parent(v) = u$

**if**  $v \in G$  **return**  $v$

        Inserta  $v$  a *Open*

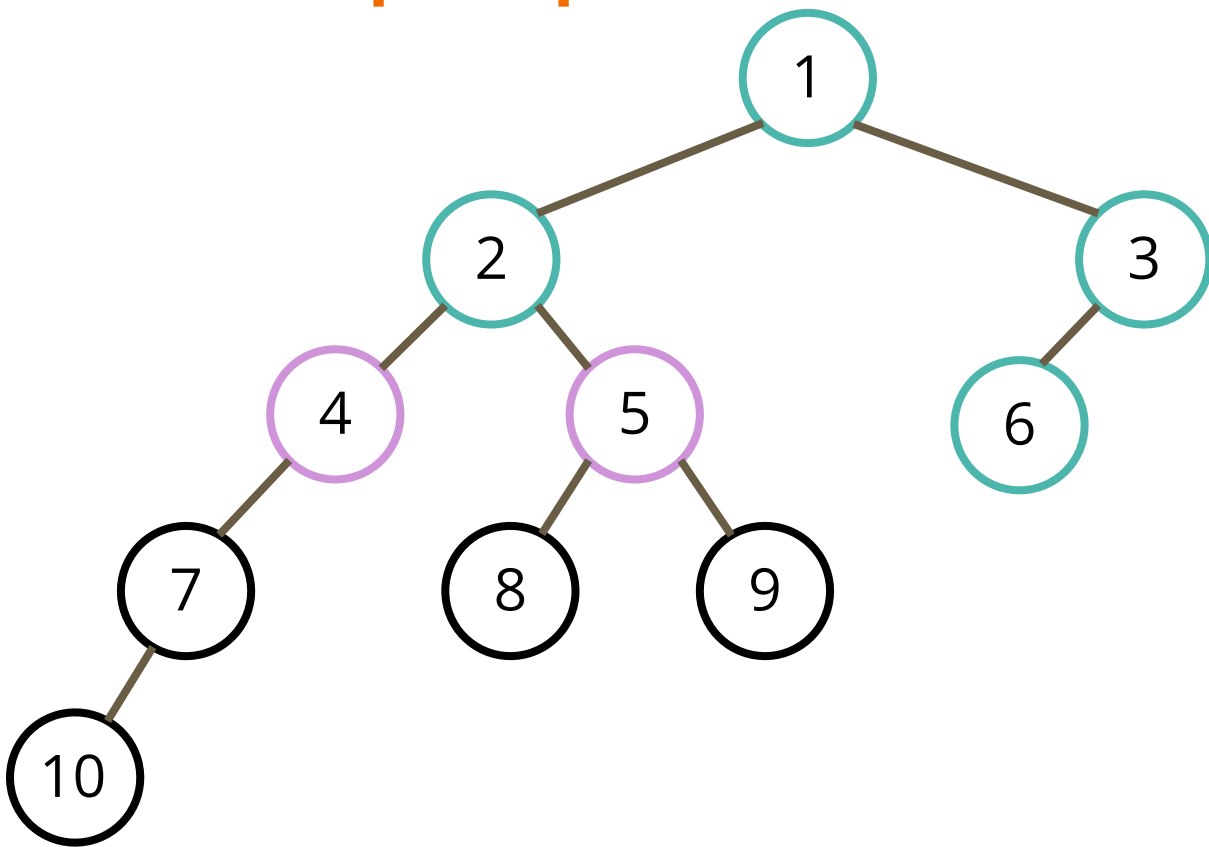
**Closed:** [1, 3, 6]

**Open:** [2]

**Goal:** {9}



# DFS - Búsqueda por Profundidad



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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

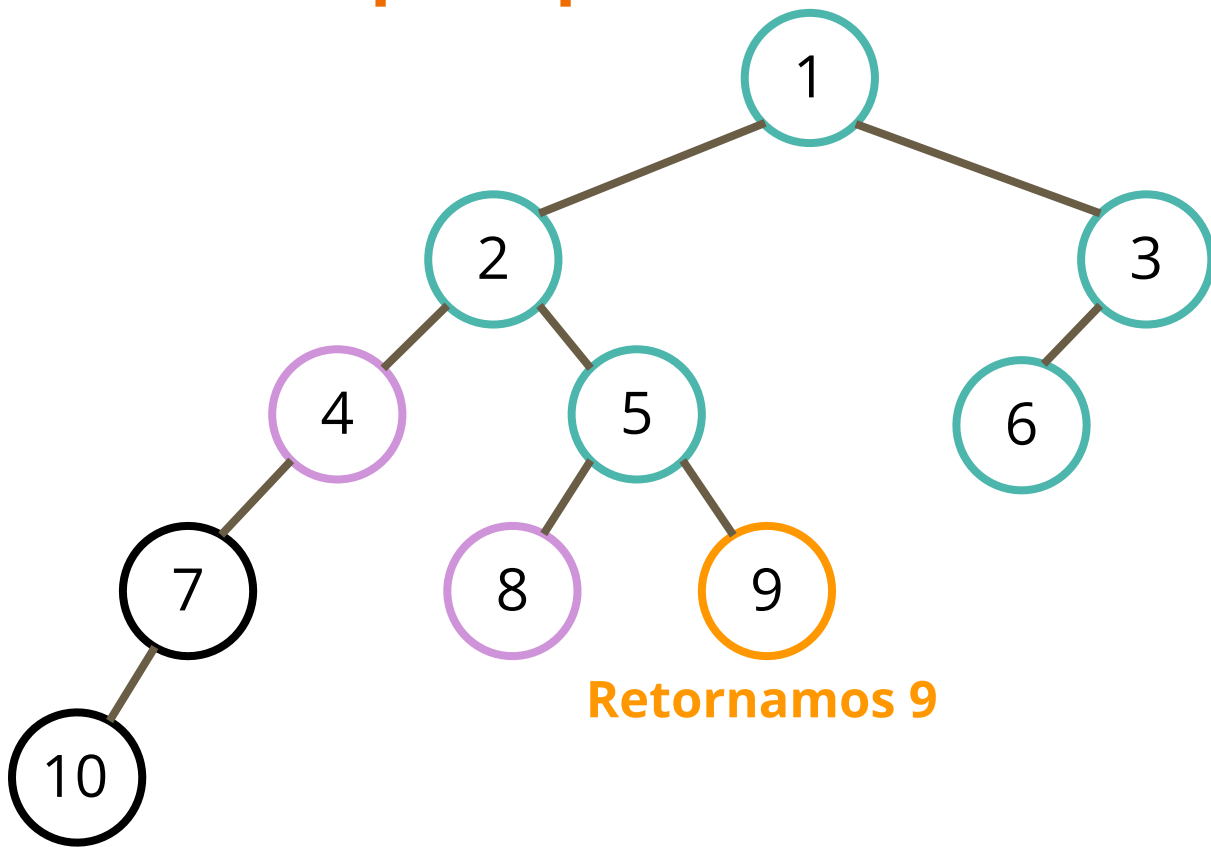
        Inserta  $v$  a *Open*

**Closed:** [1, 3, 6, 2]

**Open:** [4, 5]

**Goal:** {9}

# DFS - Búsqueda por Profundidad



Retornamos 9

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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

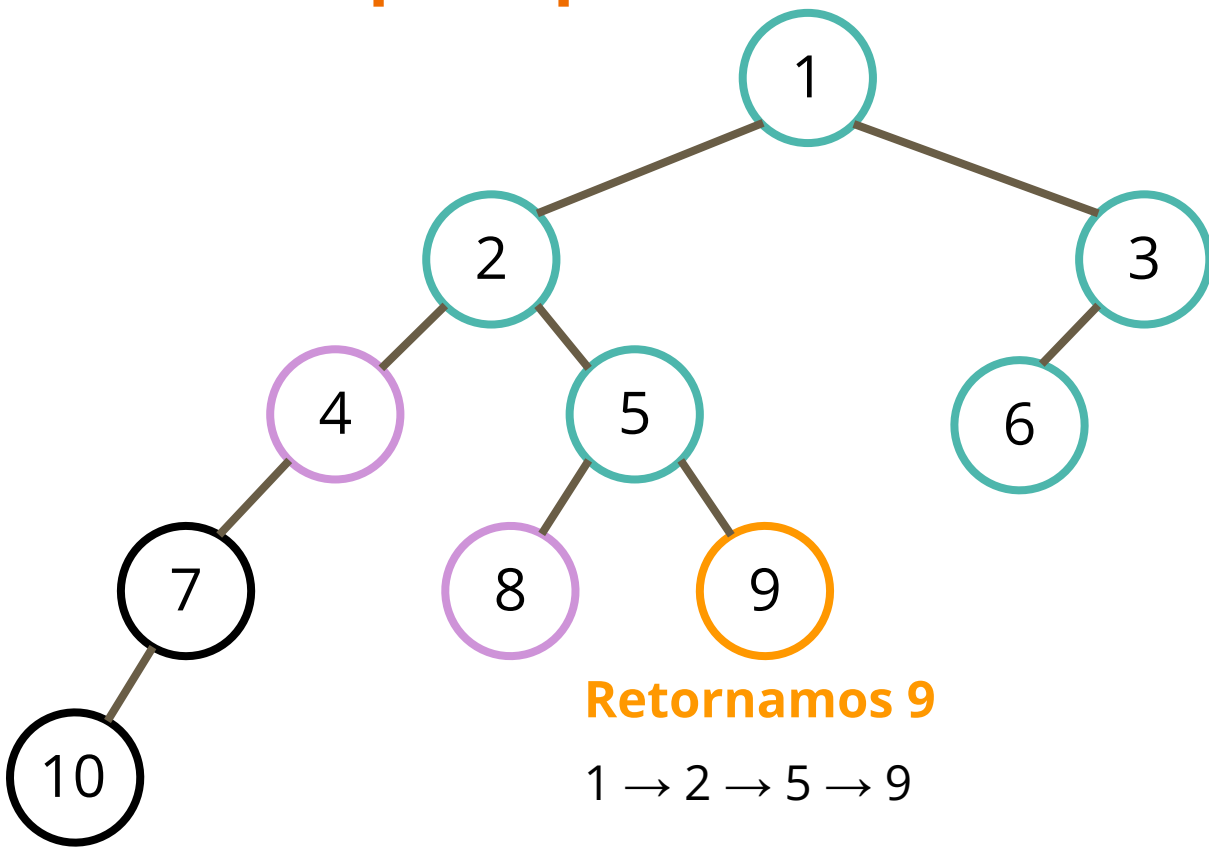
        Inserta  $v$  a *Open*

**Closed:** [1, 3, 6, 2, 5]

**Open:** [4, 8]

**Goal:** {9}

# DFS - Búsqueda por Profundidad



**Retornamos 9**

$1 \rightarrow 2 \rightarrow 5 \rightarrow 9$

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*

$\text{parent}(s_{init}) = \text{null}$

**while** *Open*  $\neq \emptyset$ :

$u \leftarrow \text{Extraer}(\text{Open})$

    Inserta  $u$  en *Closed*

**for each**  $v \in \text{Succ}(u) \setminus (\text{Open} \cup \text{Closed})$

$\text{parent}(v) = u$

**if**  $v \in G$  **return**  $v$

        Inserta  $v$  a *Open*

**Closed:** [1, 3, 6, 2, 5]

**Open:** [4, 8]

**Goal:** {9}

# IDDFS

Búsqueda por Profundidad Iterativa

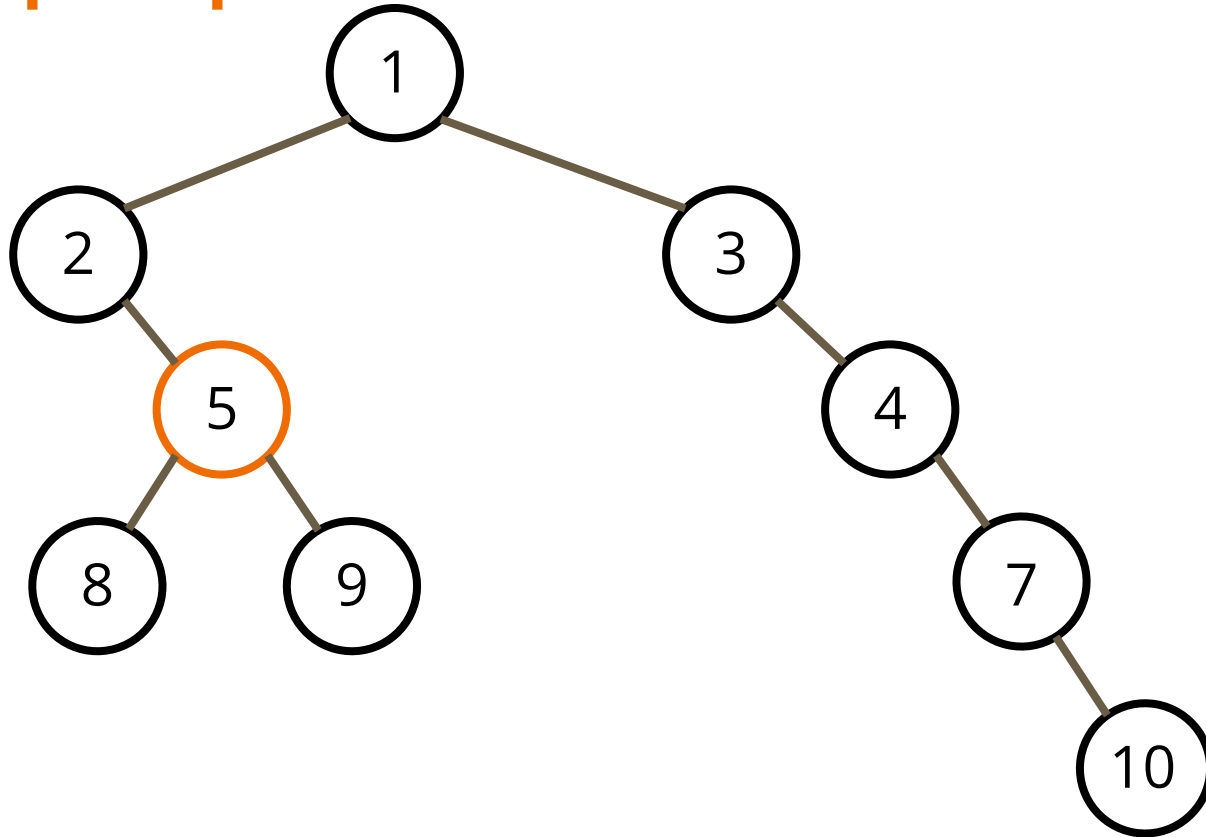
# IDDFS - Búsqueda por Profundidad Iterativa

- **b**: factor de ramificación promedio.
- **p**: profundidad a la que se encuentra la solución.
- **m**: largo de la rama más larga del árbol de búsqueda.
- **L**: profundidad máxima de la iteración.

## Pasos:

1.  $L=1$
2. Realizar búsqueda en profundidad limitada con límite  $L$ .
3. Si hubo éxito, retorna el estado encontrado; en otro caso incrementar  $L$  y volver al paso anterior.

# IDDFS - Búsqueda por Profundidad Iterativa



# Dijkstra

Algoritmo de Caminos Mínimos

# Dijkstra

