

# IIC2685 Robótica Móvil I – 2022

## Capítulo 5

### Planeamiento de Rutas

Profesor: Gabriel Sepúlveda V.  
grsepulveda@ing.puc.cl

# Agenda

- Entender el problema general de *Path Planning*
- Cell Decomposition
- Problema de búsqueda en grafos

# Problema general de Path Planning

Recordemos las **Grandes Preguntas** de un robot móvil

*¿Dónde estoy?*

*¿Para dónde voy?*

*¿Cómo llego?*



# Problema general de Path Planning

Recordemos las **Grandes Preguntas** de un robot móvil

*¿Dónde estoy?*

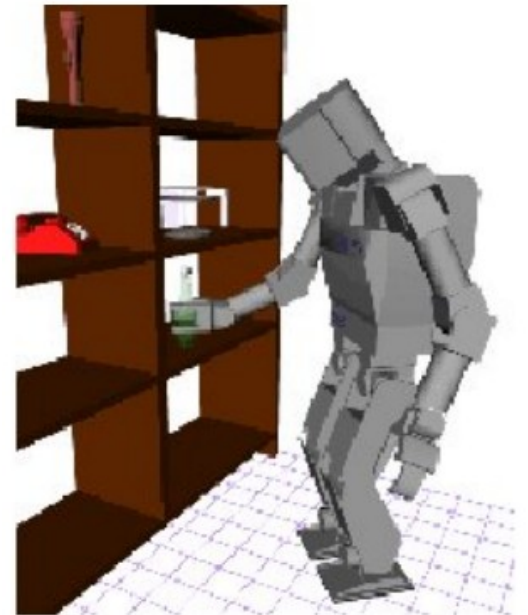
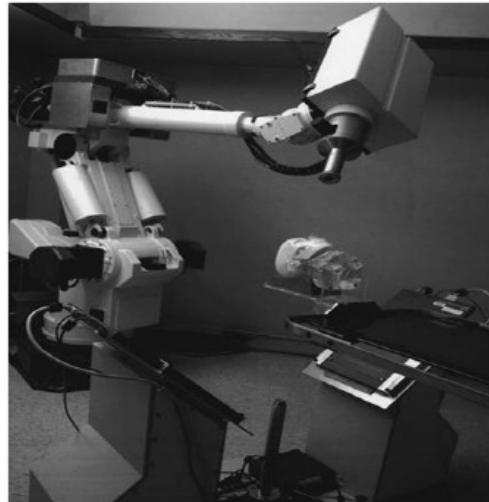
*¿Para dónde voy?*

*¿Cómo llego?*



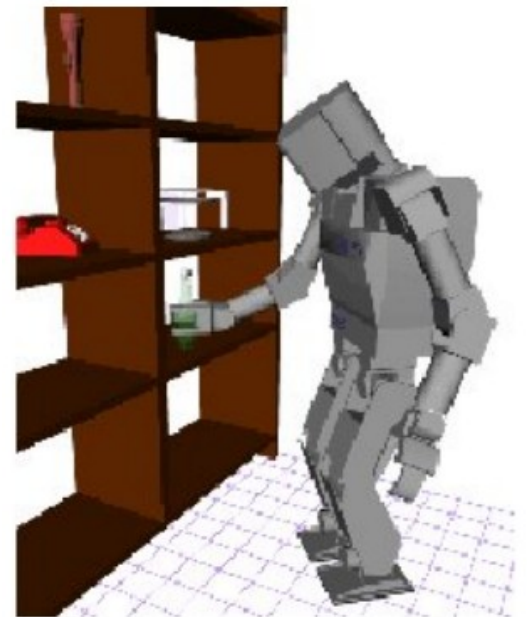
# Problema general de Path Planning

- Esta es una de las tareas fundamentales de un robot móvil
- Un robot debe tomar una **serie de decisiones** o seguir una **serie de acciones** para alcanzar su objetivo
- En general, estos corresponden a objetivos macro o de alto nivel como:
  - Recargar batería
  - Abrir un puerta
  - Alcanzar una pose objetivo



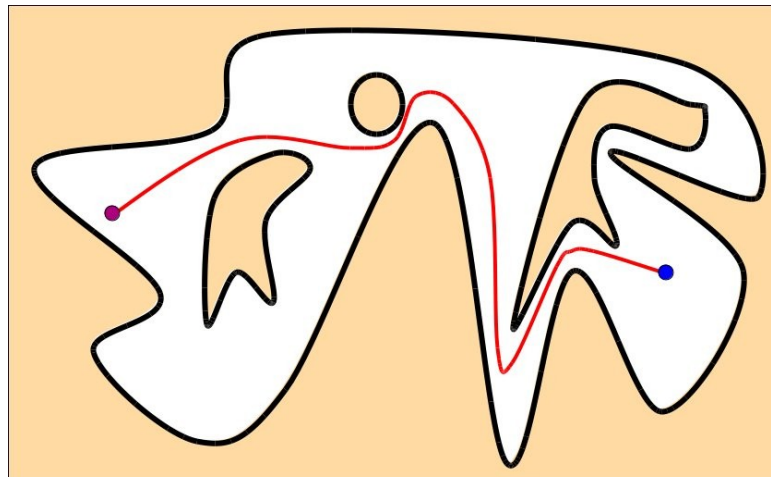
# Problema general de Path Planning

- Esta es una de las tareas fundamentales de un robot móvil
- Un robot debe tomar una **serie de decisiones** o seguir una **serie de acciones** para alcanzar su objetivo
- En general, estos corresponden a objetivos macro o de alto nivel como:
  - Recargar batería
  - Abrir un puerta
  - **Alcanzar una pose objetivo**



# Path Planning

- Para nuestro problema de navegación, a partir de un punto de origen y uno de destino:
  - Buscaremos la secuencia de **estados** que conecten estos puntos de la mejor forma posible
  - Estos estados serán representados por las **poses** en el plano 2D
  - Finalmente, escogeremos la secuencia de **acciones** que nos permitan transitar entre estados sucesivos



# Global Path Planning

- **Objetivo:** explorar el **espacio libre** hasta obtener una ruta válida desde la posición de origen hasta la posición objetivo



# Global Path Planning

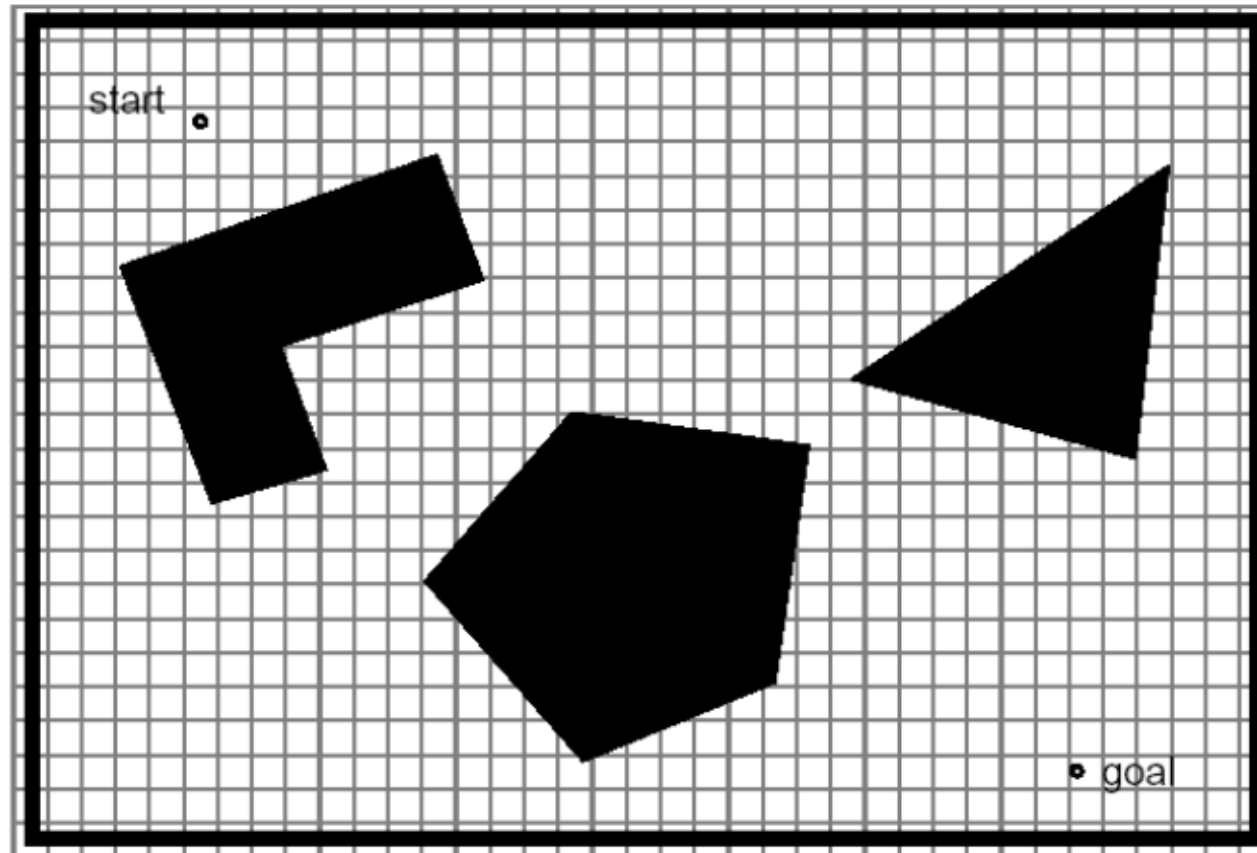
- **Objetivo:** explorar el **espacio libre** hasta obtener una ruta válida desde la posición de origen hasta la posición objetivo
- Existen varias aproximaciones clásicas:
  - Roadmap: red de curvas de conectividad en 1-D
  - Potential Fields: suma de campos de atracción ( objetivo ) y repulsión ( obstáculos )
  - Cell Decomposition: descomposición de mapa en regiones/celdas

# Global Path Planning

- **Objetivo:** explorar el **espacio libre** hasta obtener una ruta válida desde la posición de origen hasta la posición objetivo
- Existen varias aproximaciones clásicas:
  - Roadmap: red de curvas de conectividad en 1-D
  - Potential Fields: suma de campos de atracción ( objetivo ) y repulsión ( obstáculos )
  - Cell Decomposition: descomposición de mapa en regiones
- En esta clase nos enfocaremos en una de las más populares: *Cell Decomposition*

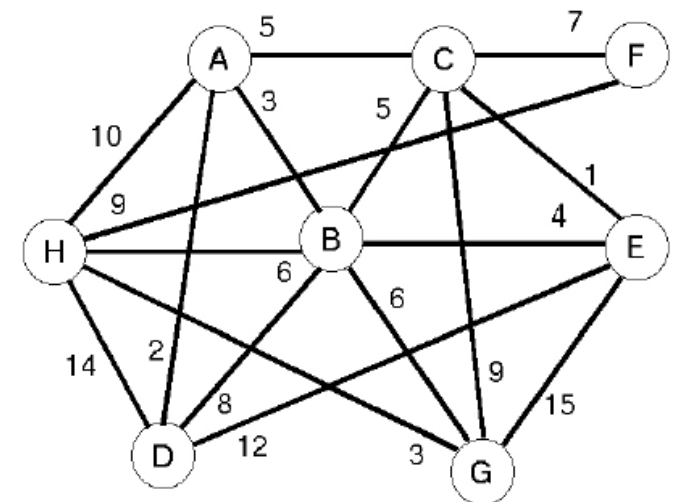
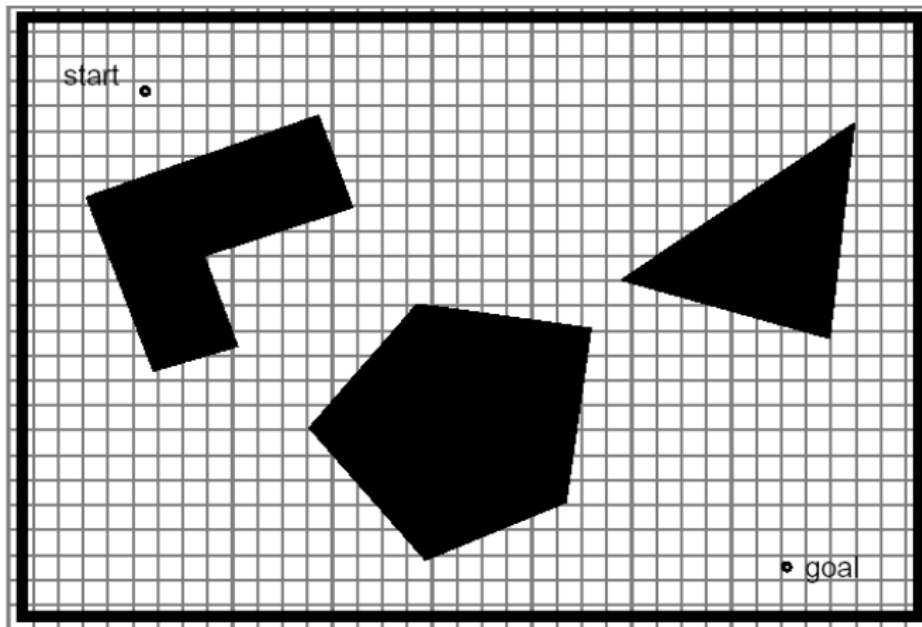
# Cell Decomposition

- Descompone el espacio en celdas ...



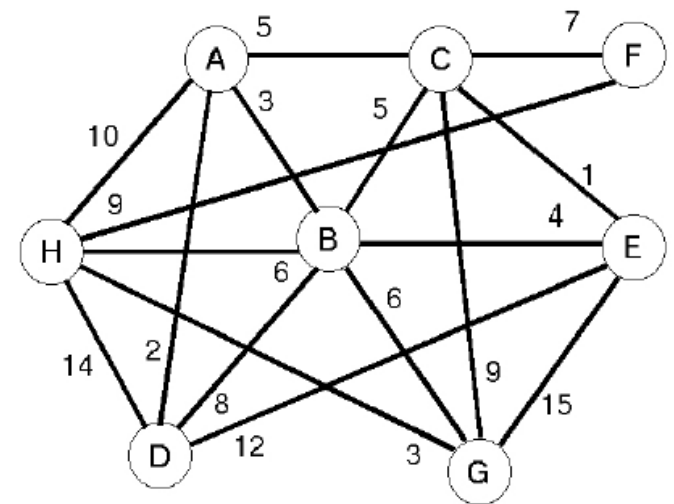
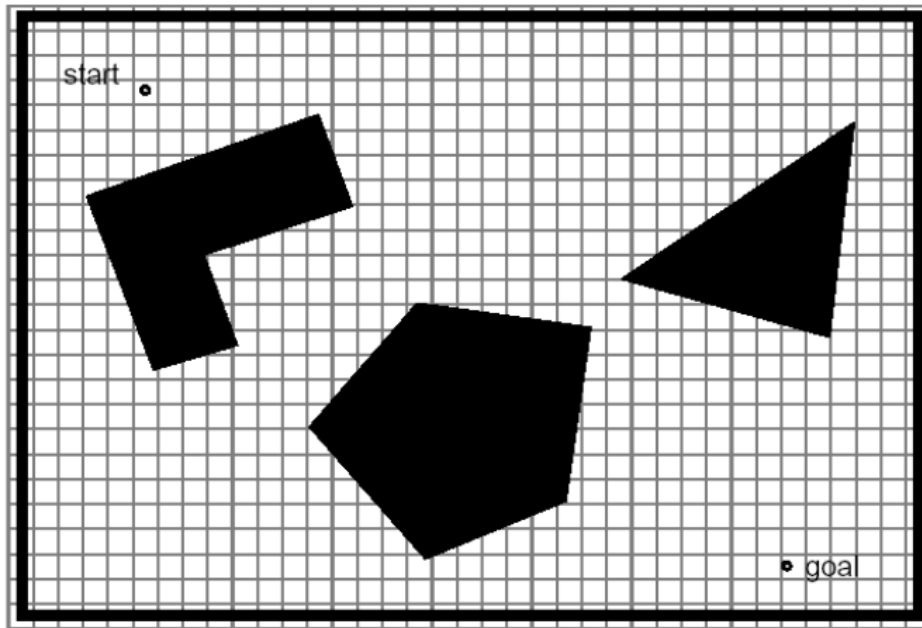
# Cell Decomposition

- Descompone el espacio en celdas y representa su conectividad a través de grafos de adyacencia



# Cell Decomposition

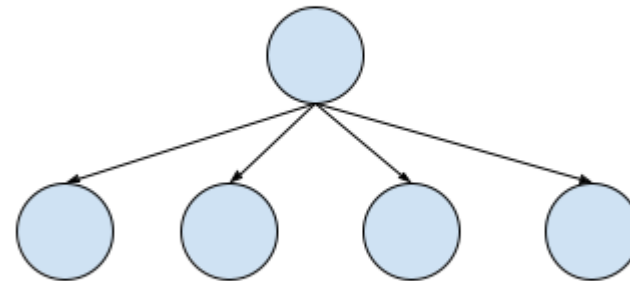
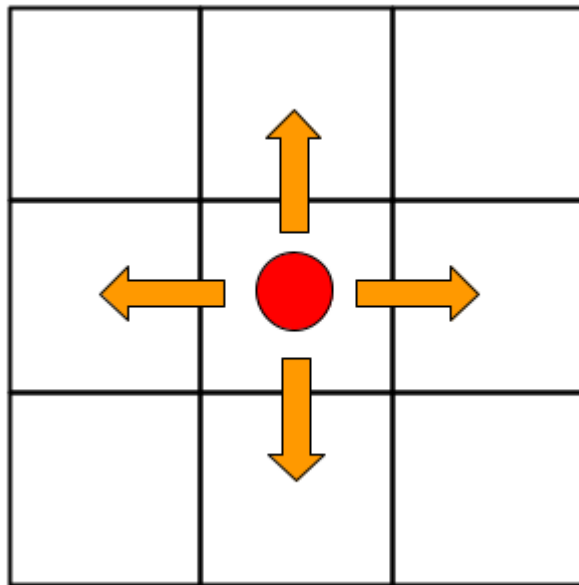
- Descompone el espacio en celdas y representa su conectividad a través de grafos de adyacencia



- ¿ Cómo construir el grafo a partir de esta representación ?

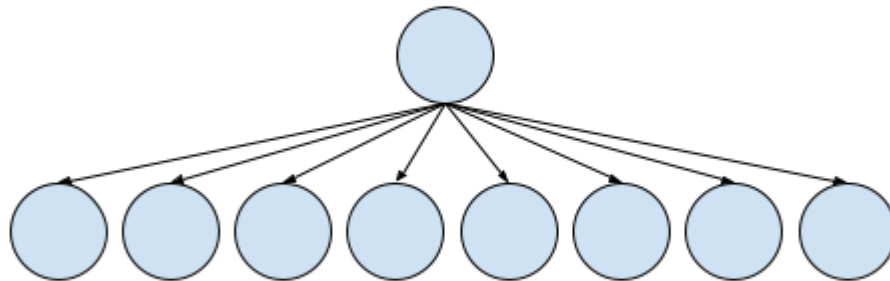
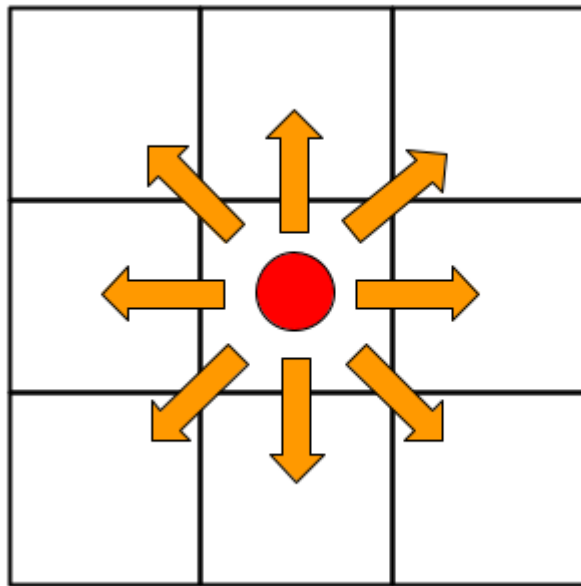
# Cell Decomposition

- Esquema **cuatro conectado**: asume que estados posibles se encuentran solo en posiciones horizontales y verticales



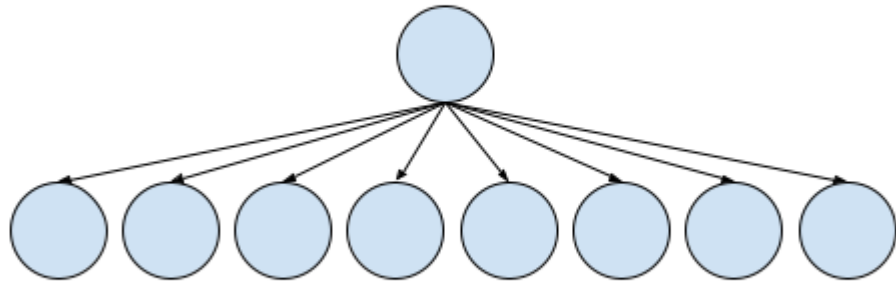
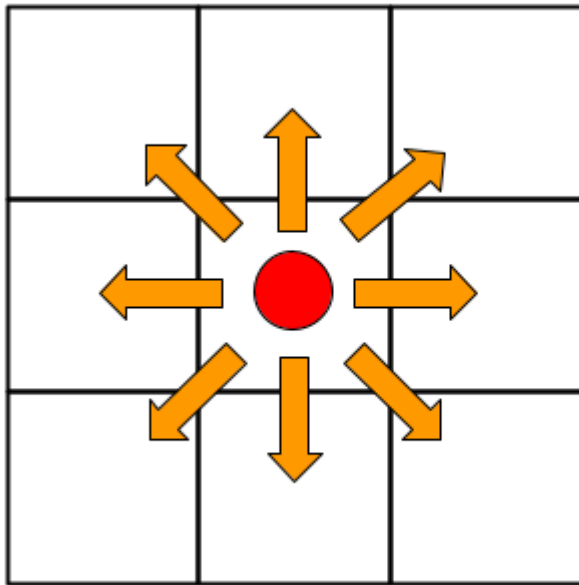
# Cell Decomposition

- Esquema **ocho conectado**: asume que estados posibles se encuentran en todas las posiciones adjacentes



# Cell Decomposition

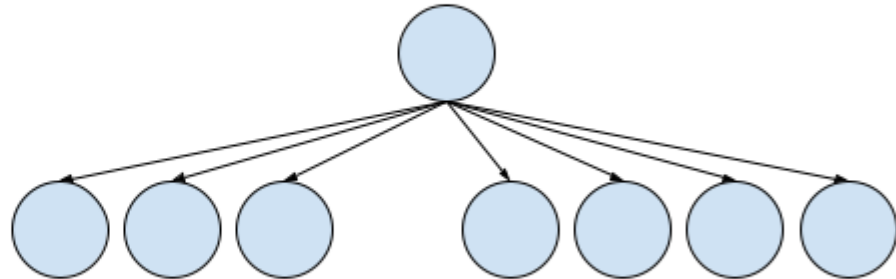
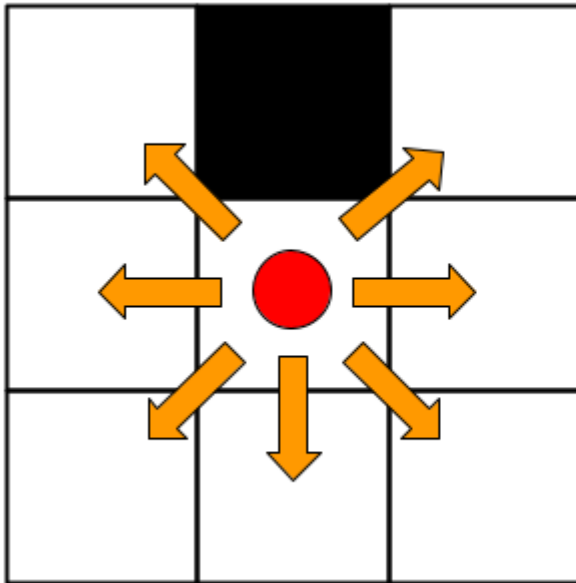
- Esquema **ocho conectado**: asume que estados posibles se encuentran en todas las posiciones adjacentes
- ¿ Cómo construir el grafo si una de la celdas está ocupada ?





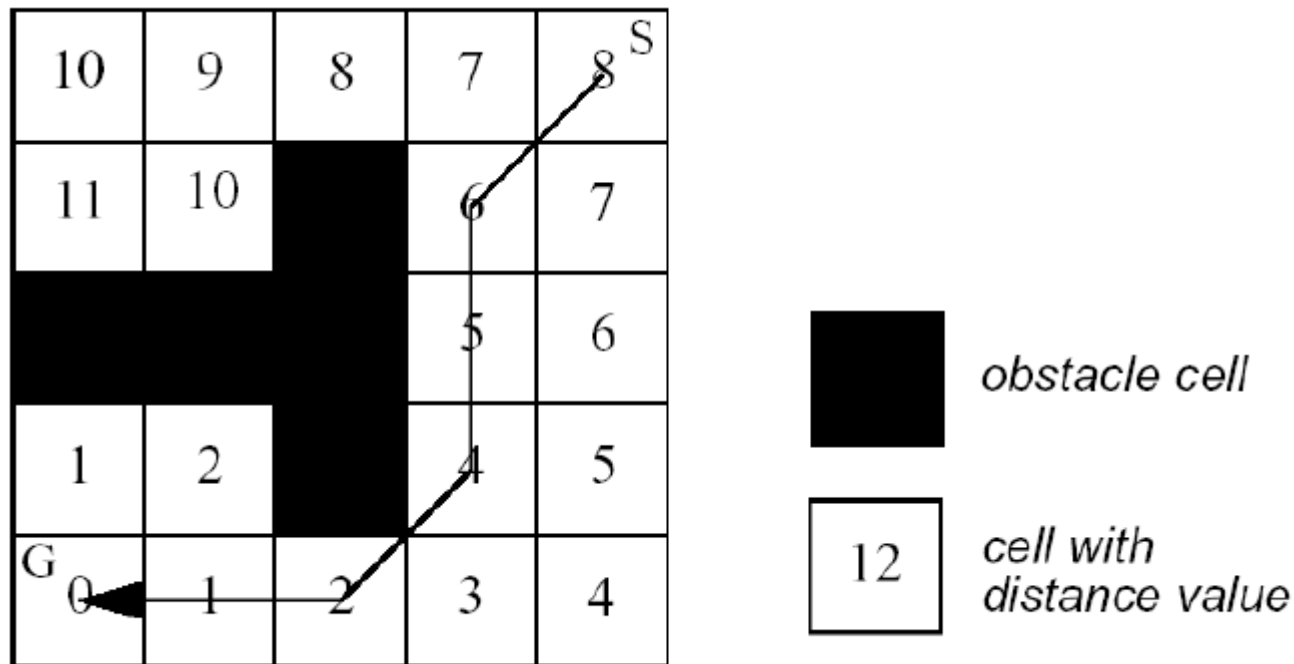
# Cell Decomposition

- Esquema **ocho conectado**: asume que estados posibles se encuentran en todas las posiciones adjacentes
- ¿Cómo construir el grafo si una de la celdas está ocupada ?



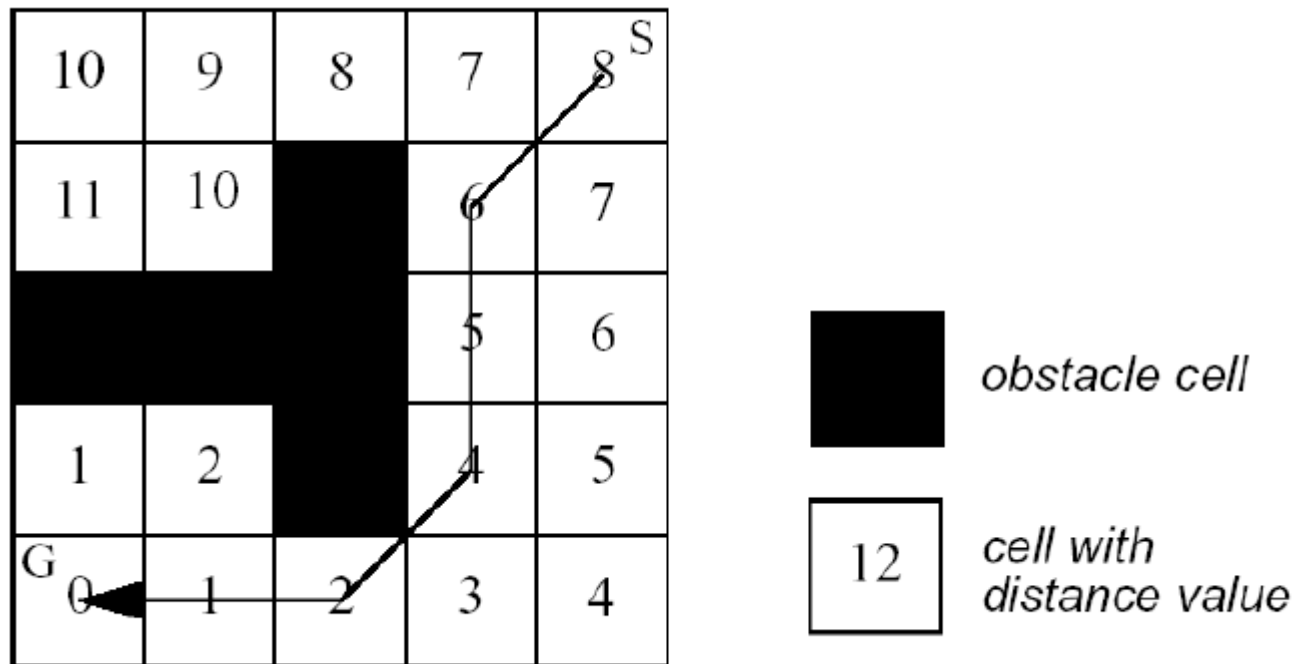
# Path Planning en Cell Decomposition

- ¿ Es posible encontrar una ruta válida utilizando algún algoritmo ?



# Path Planning en Cell Decomposition

- ¿ Es posible encontrar una ruta válida utilizando algún algoritmo ?

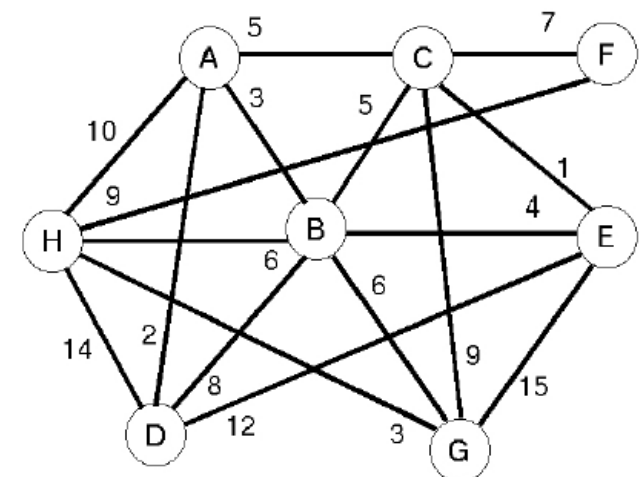
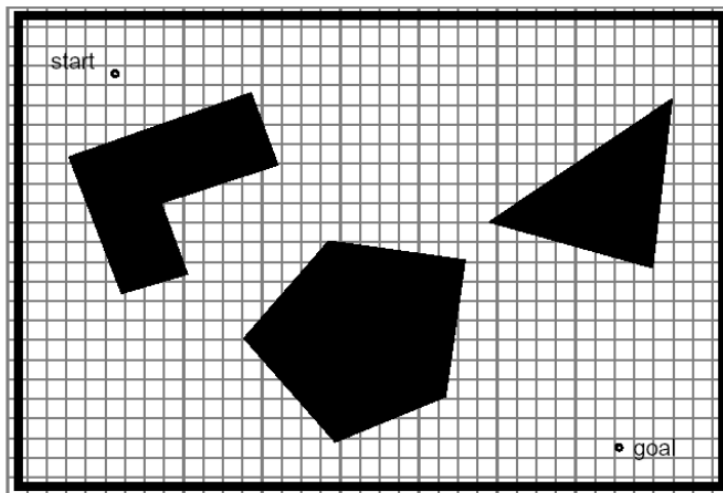


- Los algoritmos de búsqueda en grafos nos darán la respuesta !

# Path Planning en Cell Decomposition

## Procedimiento general de planeamiento de rutas

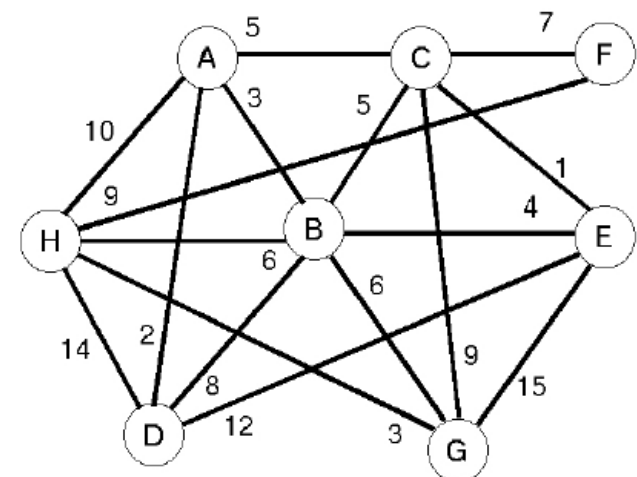
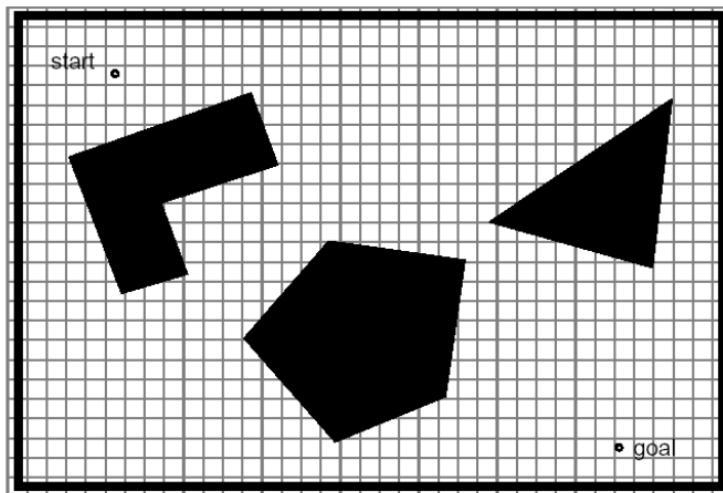
- Dividir el espacio en regiones conectadas llamadas celdas
- Determinar que celdas libres son adjacetas, y construir a partir de ellas un grafo de conectividad
  - Celda → nodo / vértice / estado
  - Conectividad → arista / arco



# Path Planning en Cell Decomposition

## Procedimiento general de planeamiento de rutas

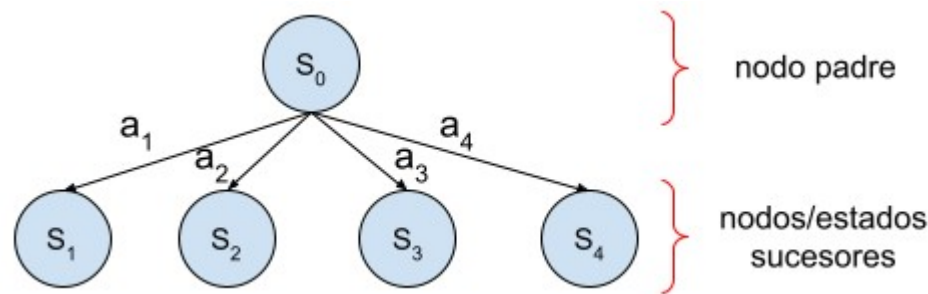
- Determinar cuáles son las celdas que contienen el punto de origen y destino, y luego determinar sus nodos equivalentes en el grafo
- **Buscar** una ruta dentro del grafo que conecte los estados ( celdas ) de origen y destino según algún criterio



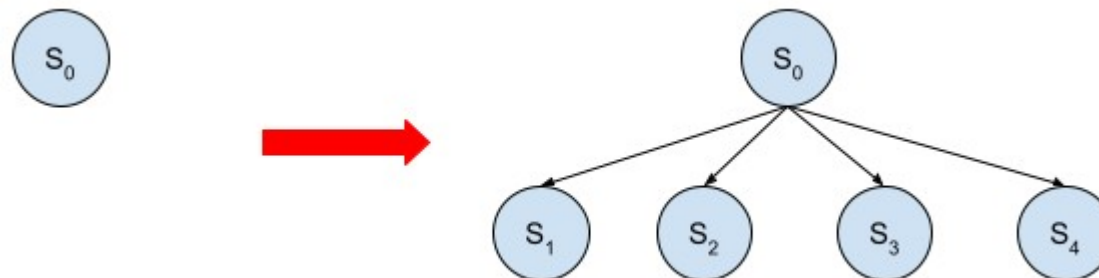
# Búsqueda en grafos

## Proceso General de Búsqueda

- La aplicación de una **acción** sobre un estado tendrá por resultado la generación de un **estado sucesor**

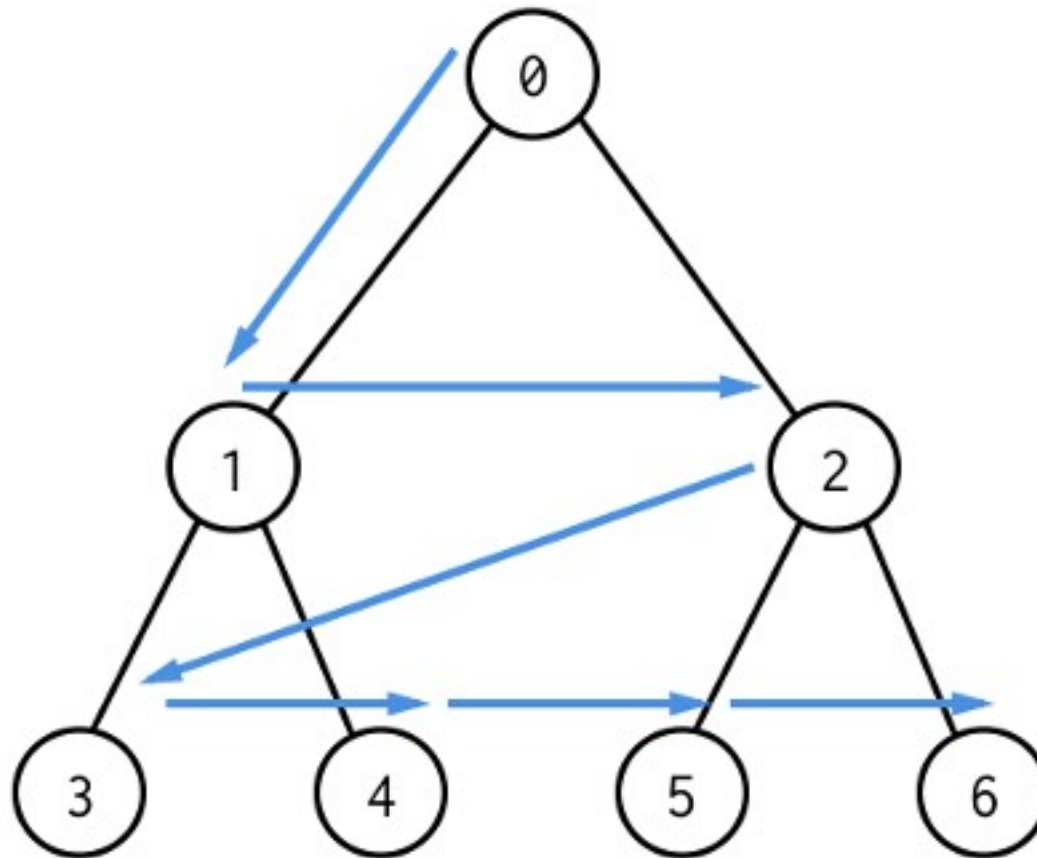


- La generación del **espacio de estados** aplicando todas las acciones posibles sobre un nodo particular se conoce como **expansión**



# Breadth First Search ( BFS )

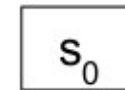
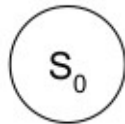
- Búsqueda en amplitud
- Orden de expansión horizontal y luego vertical



# Breadth First Search ( BFS )

**Ejemplo:** Start =  $s_0$ , Goal =  $s_8$

- Se expande estado  $s_0$  y se agrega a una cola de estados

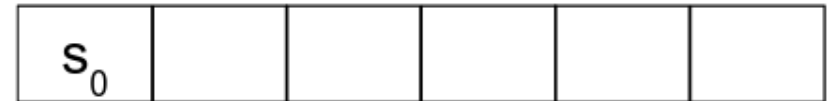




# Breadth First Search ( BFS )

**Ejemplo:** Start =  $s_0$ , Goal =  $s_8$

- Se expande estado  $s_0$  y se agrega a una cola de estados



- Se saca el estado *más antiguo* desde la cola, y se expande

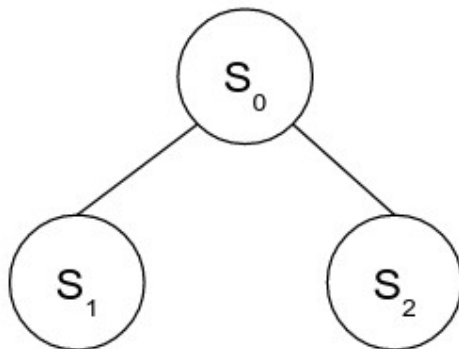
# Breadth First Search ( BFS )

**Ejemplo:** Start =  $s_0$ , Goal =  $s_8$

- Se expande estado  $s_0$  y se agrega a una cola de estados

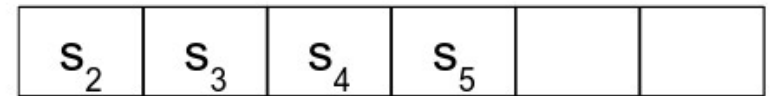
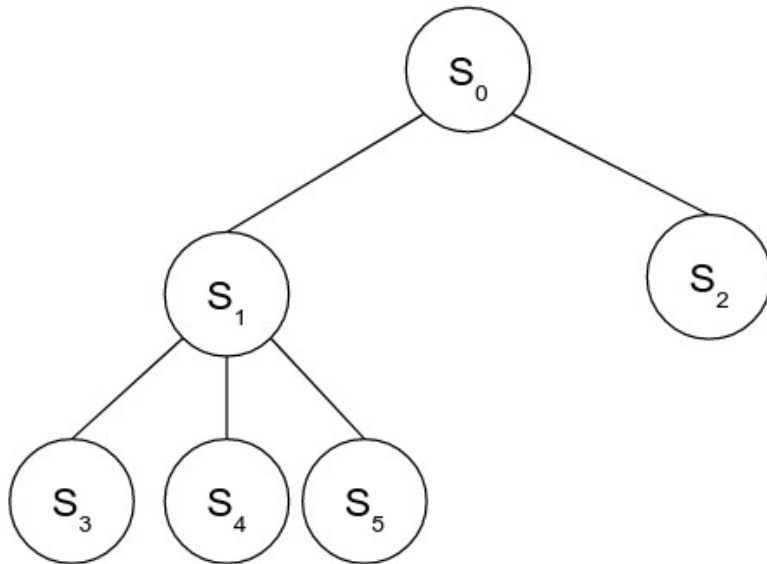


- Se saca el estado *más antiguo* desde la cola, y se expande
- Estados hijos se almacenan en la cola



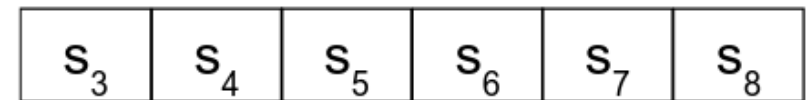
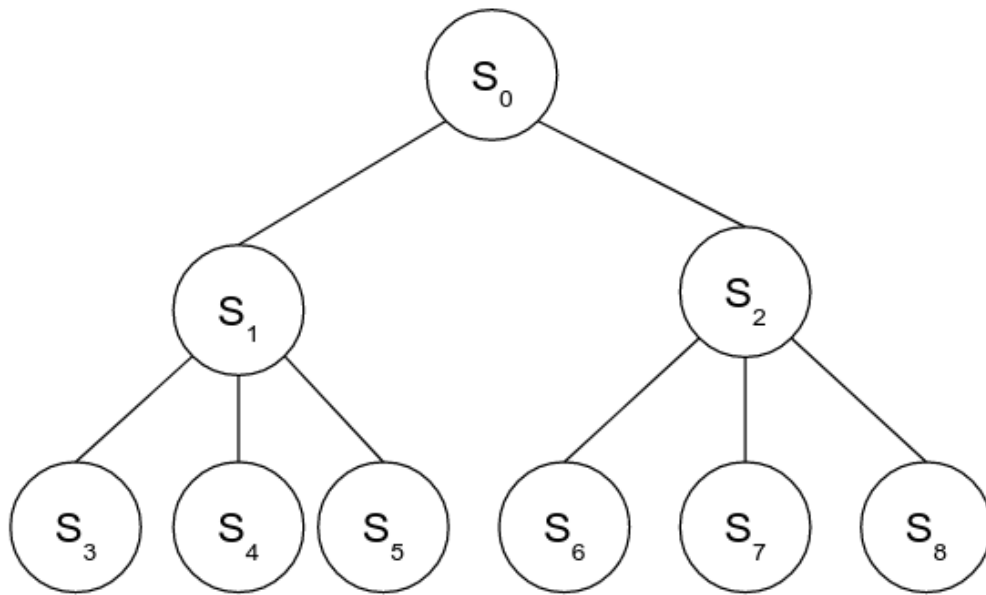
# Breadth First Search ( BFS )

- Se saca el estado *más antiguo* desde la cola, y se expande
- Estados hijos se almacenan en la cola



# Breadth First Search ( BFS )

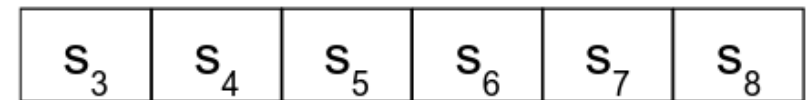
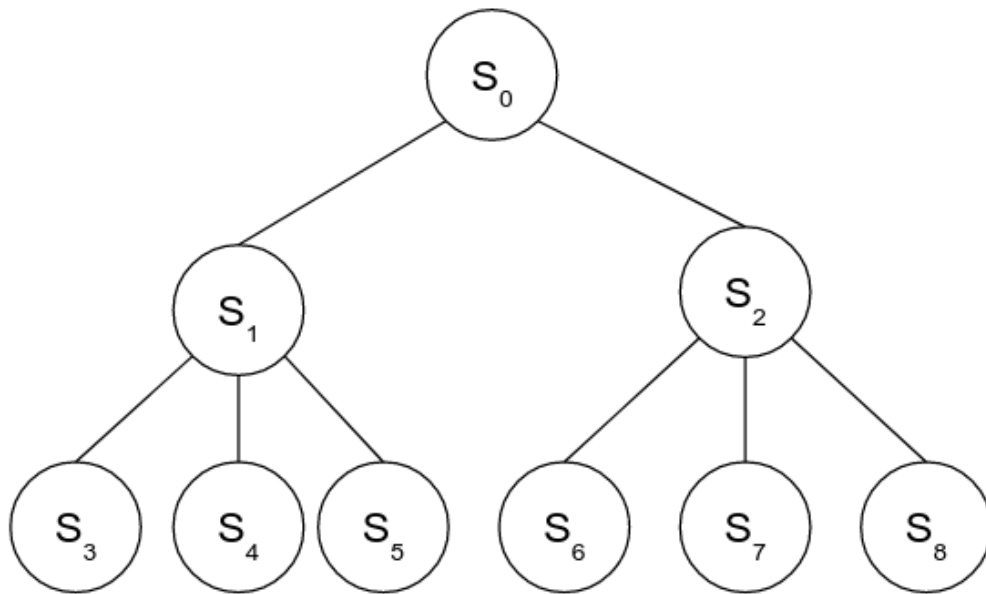
- Se saca el estado *más antiguo* desde la cola, y se expande
- Estados hijos se almacenan en la cola



- Hemos encontrado estado objetivo S<sub>8</sub> !
- Cola de expansión es del tipo ...

# Breadth First Search ( BFS )

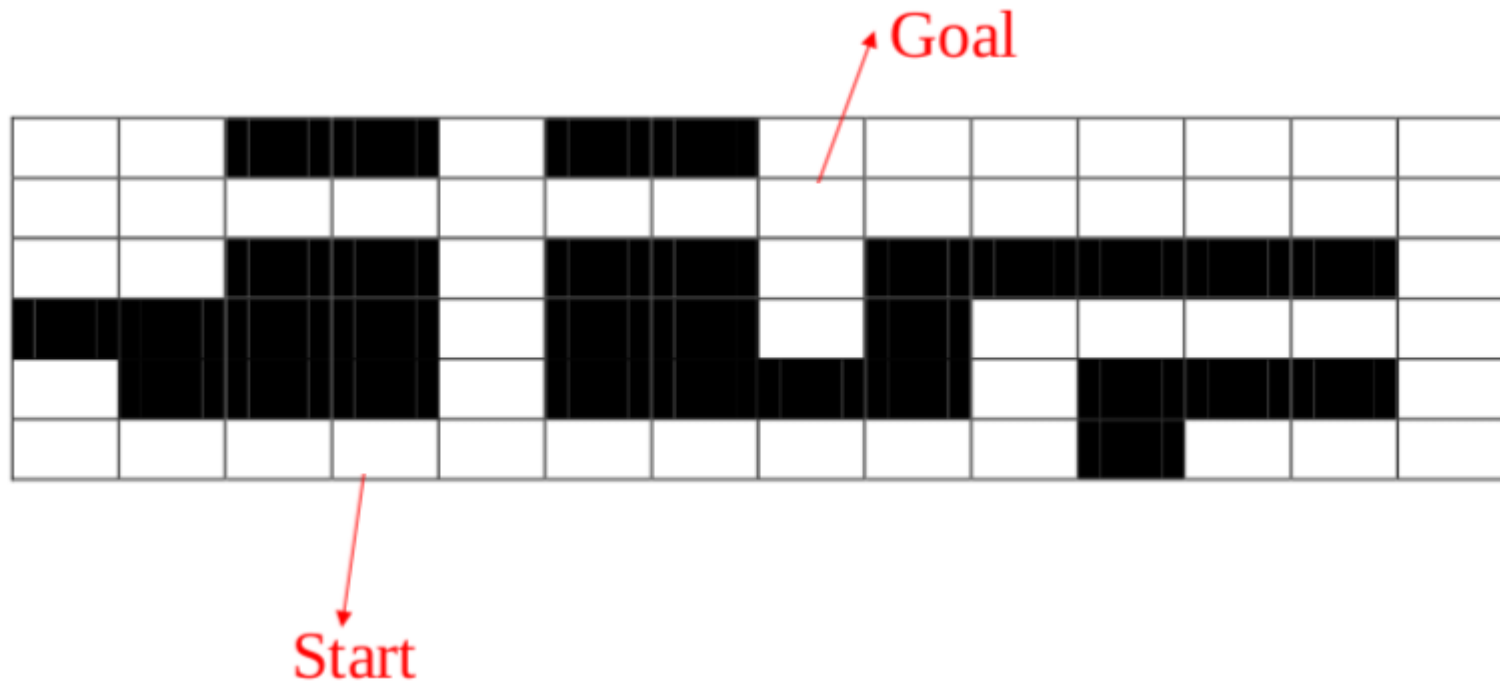
- Se saca el estado *más antiguo* desde la cola, y se expande
- Estados hijos se almacenan en la cola



- Hemos encontrado estado objetivo S<sub>8</sub> !
- Cola de expansión es del tipo **FIFO**

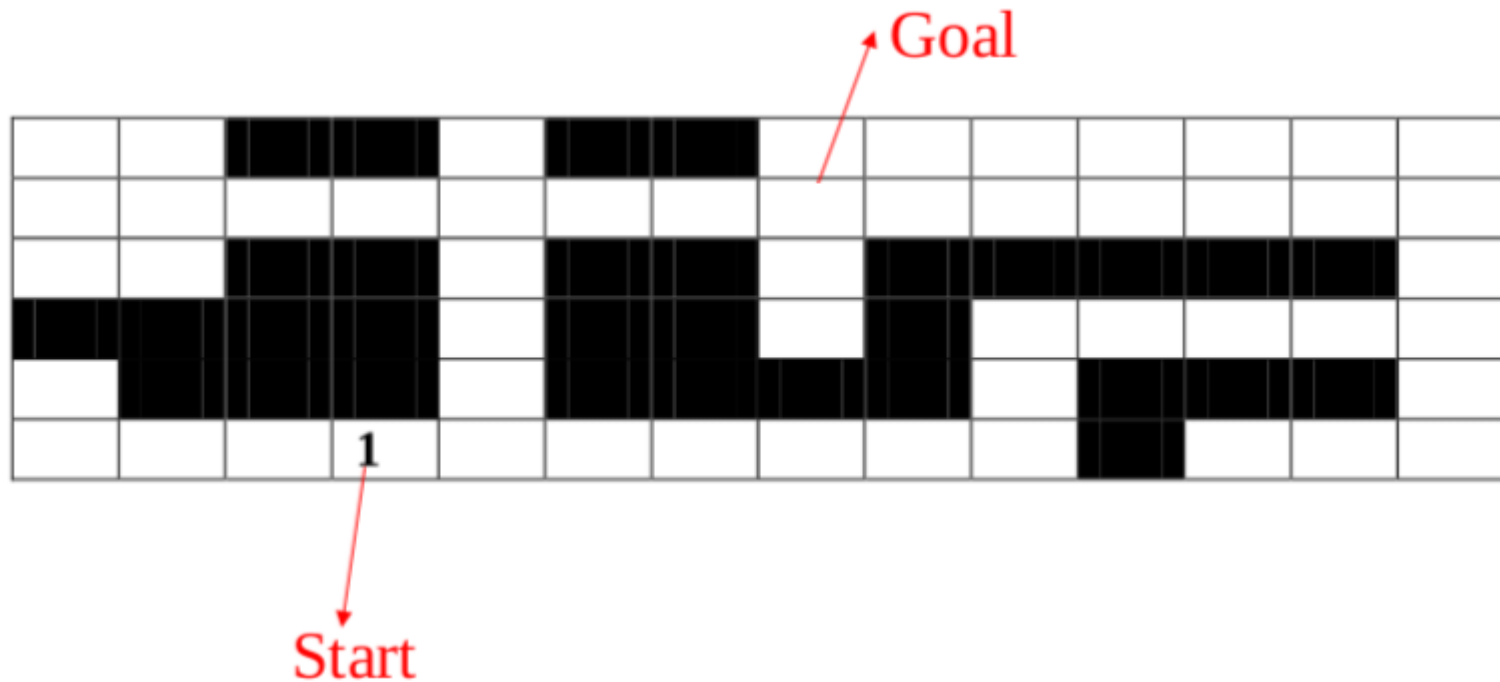
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



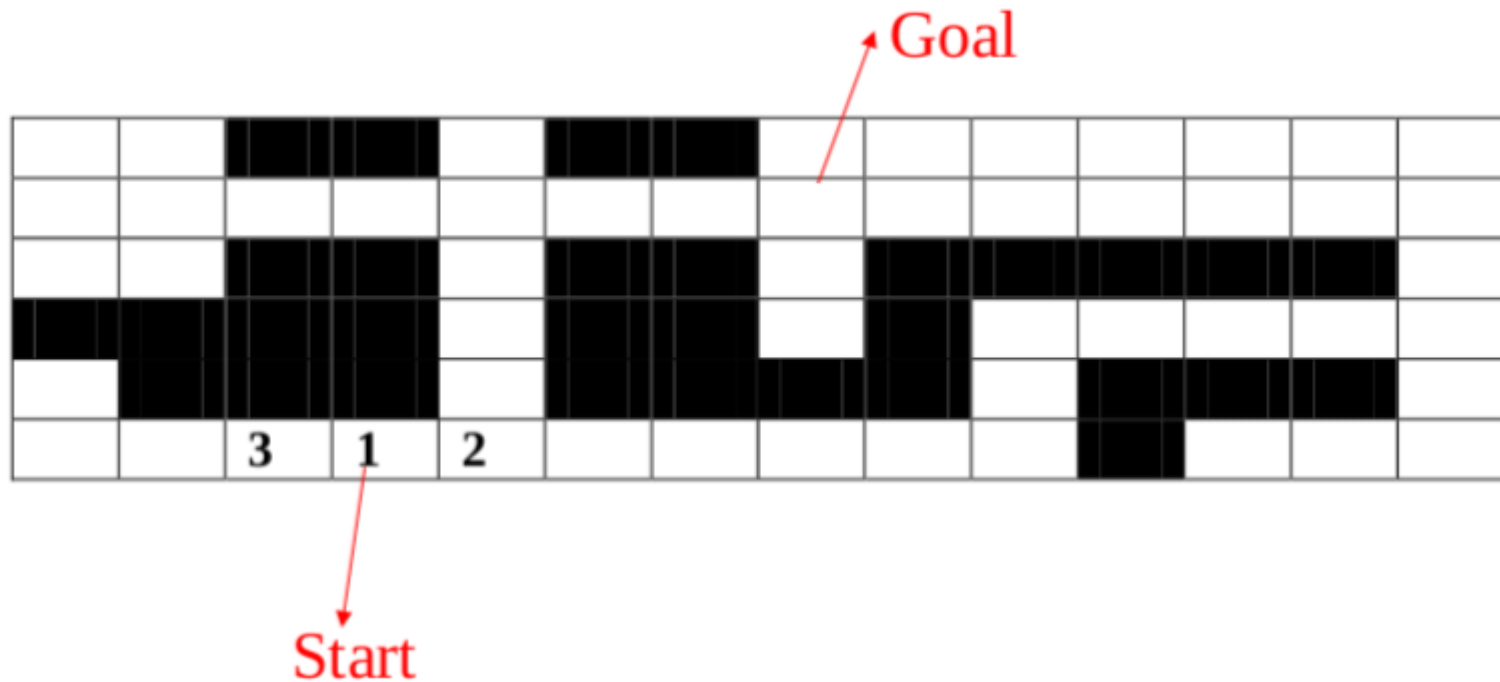
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

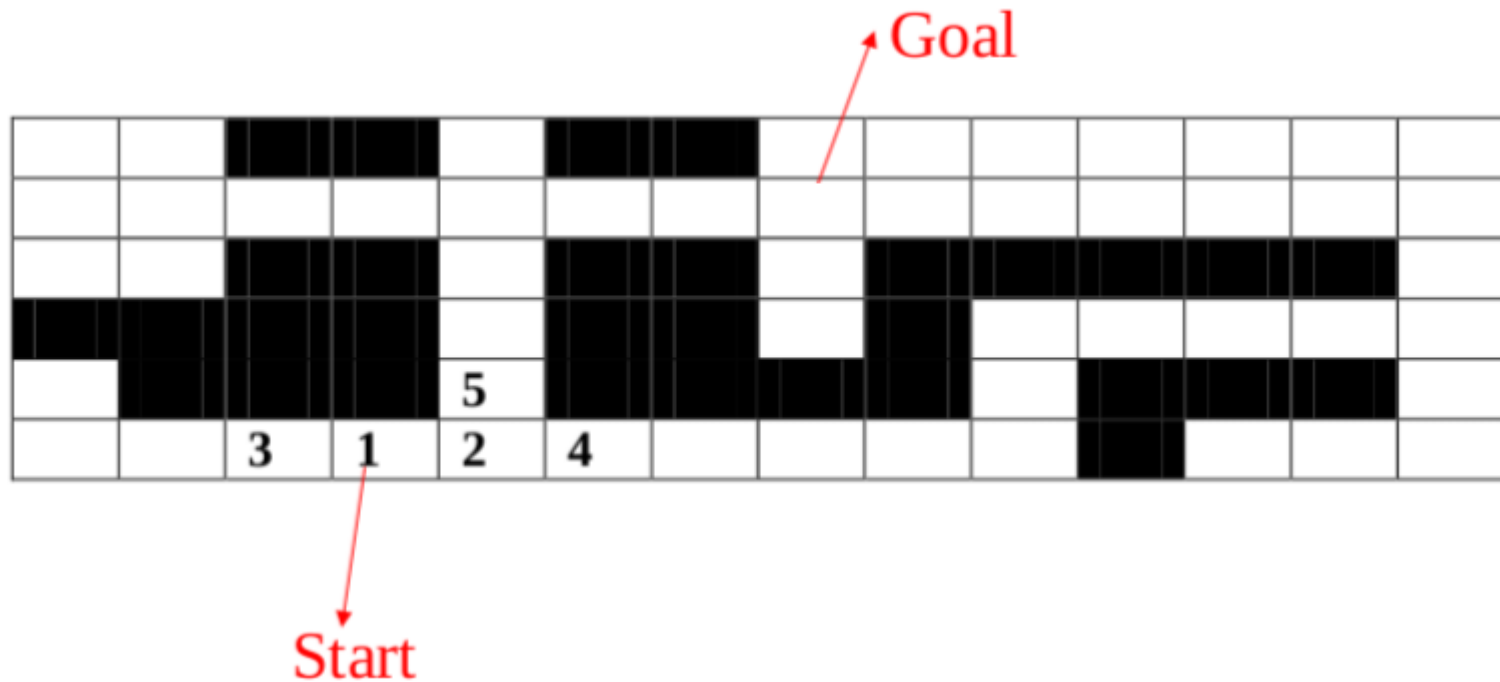
- ¿ Cómo se desarrolla BFS sobre el mapa ?





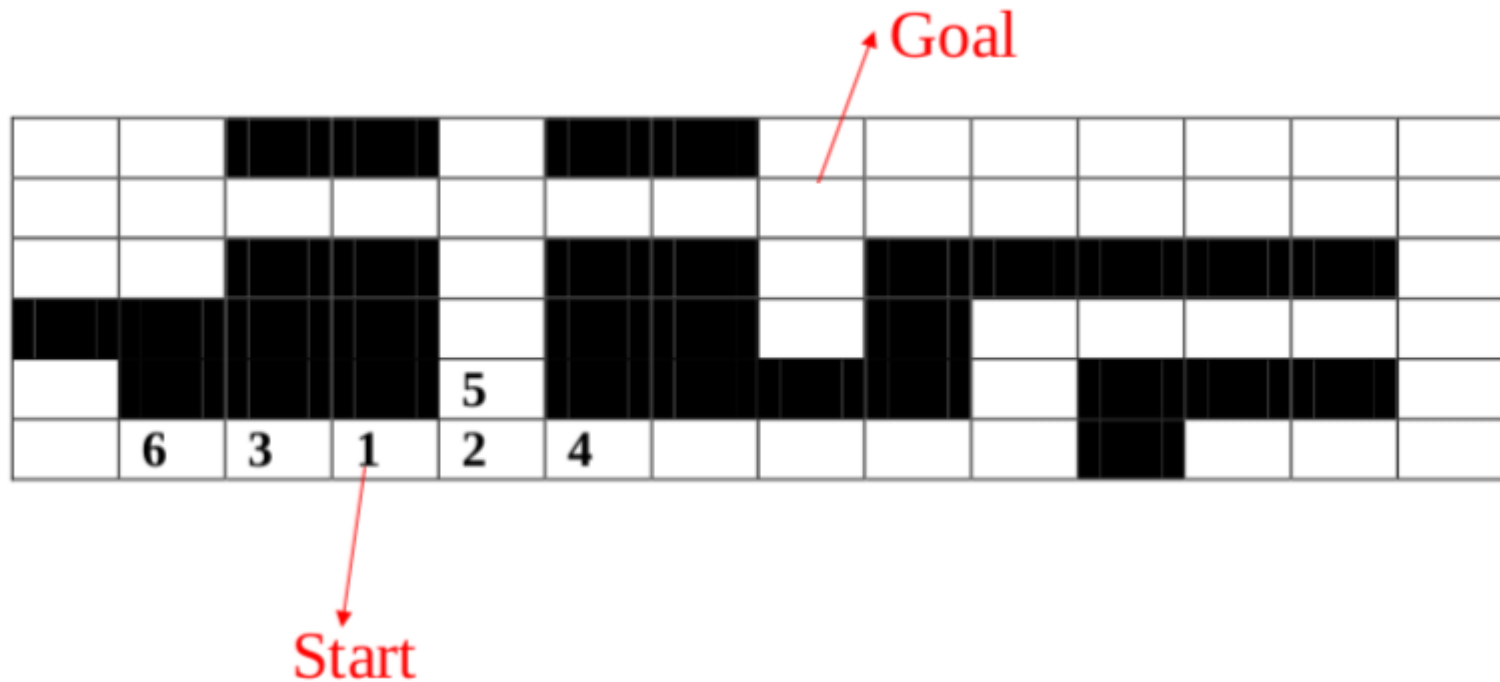
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



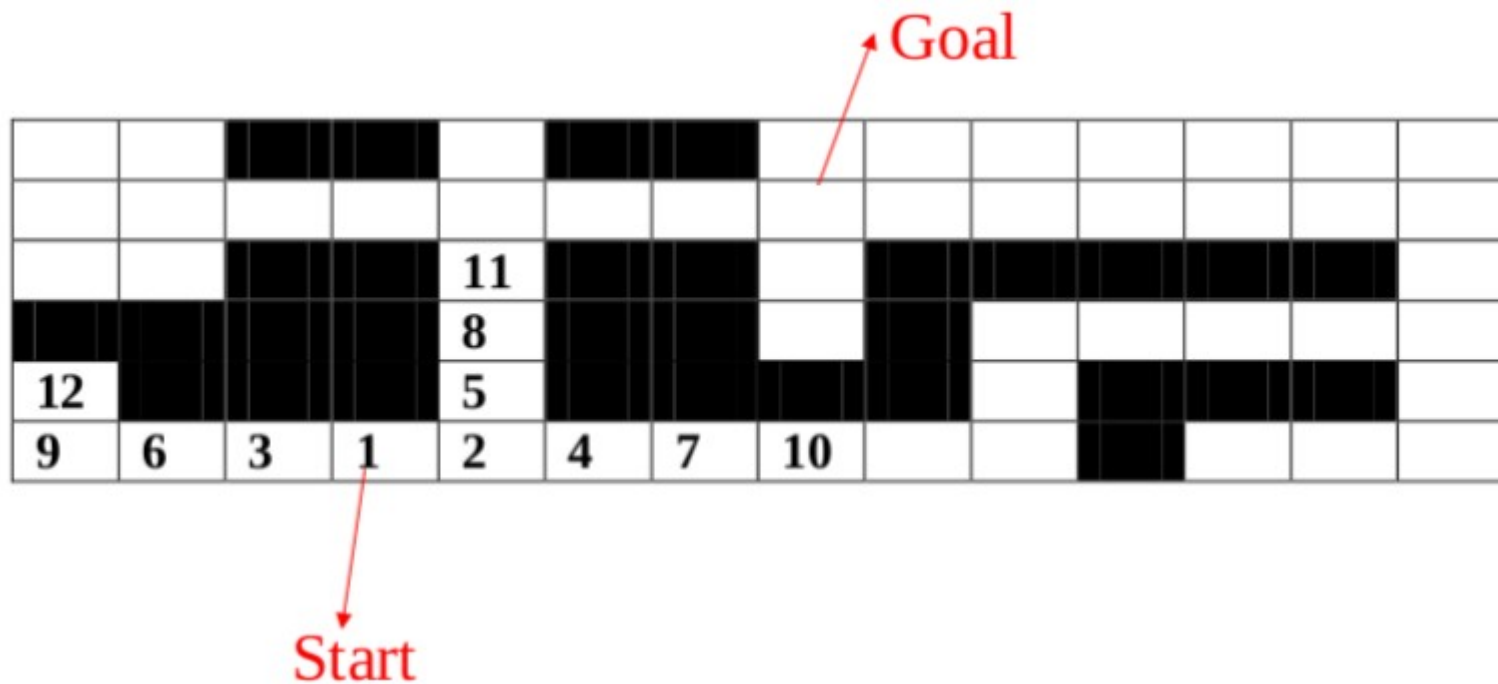
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

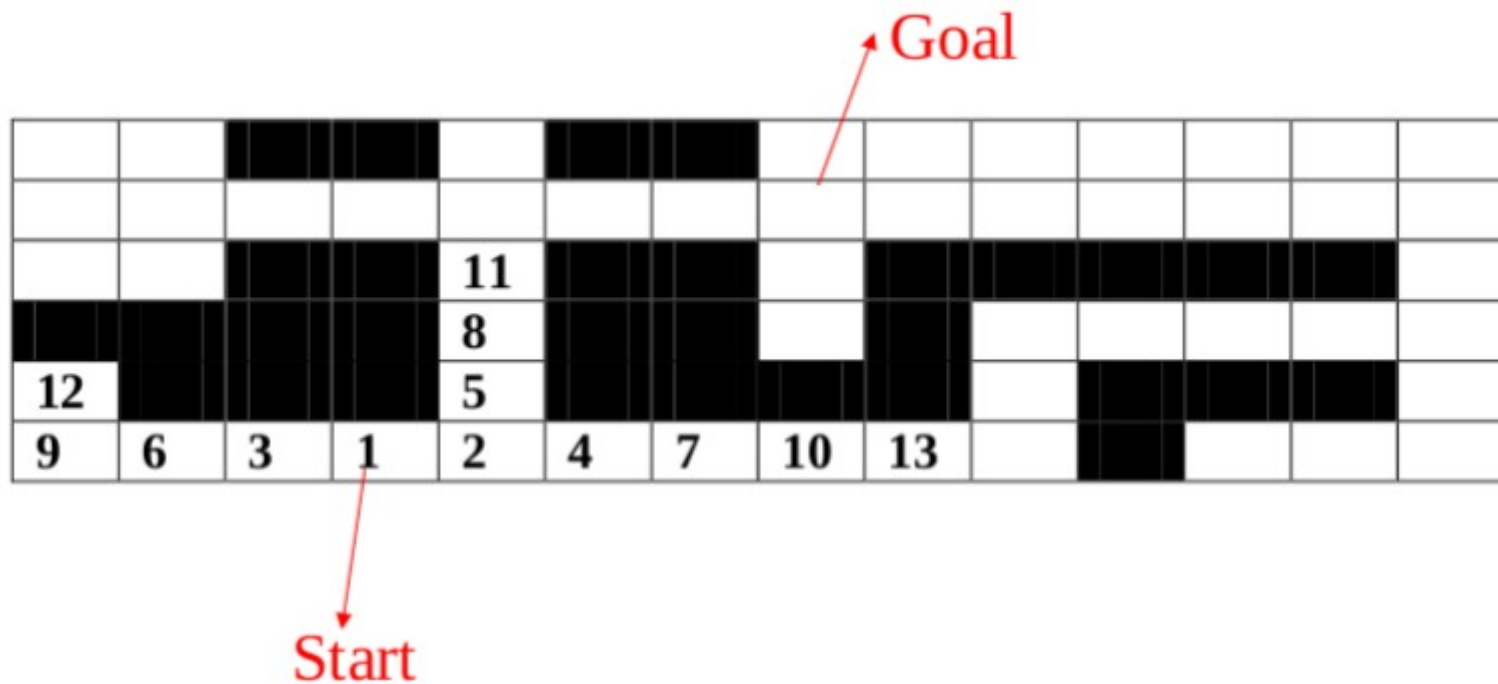
- ¿ Cómo se desarrolla BFS sobre el mapa ?





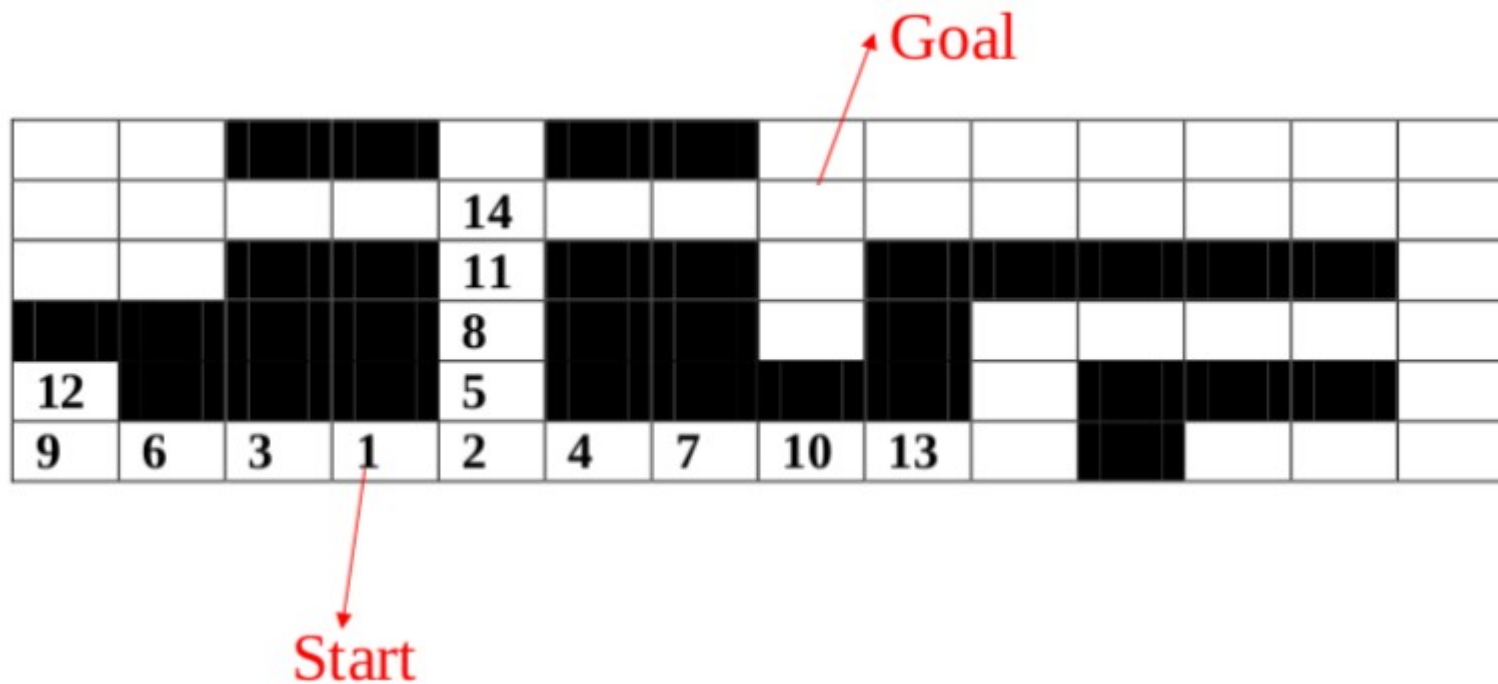
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



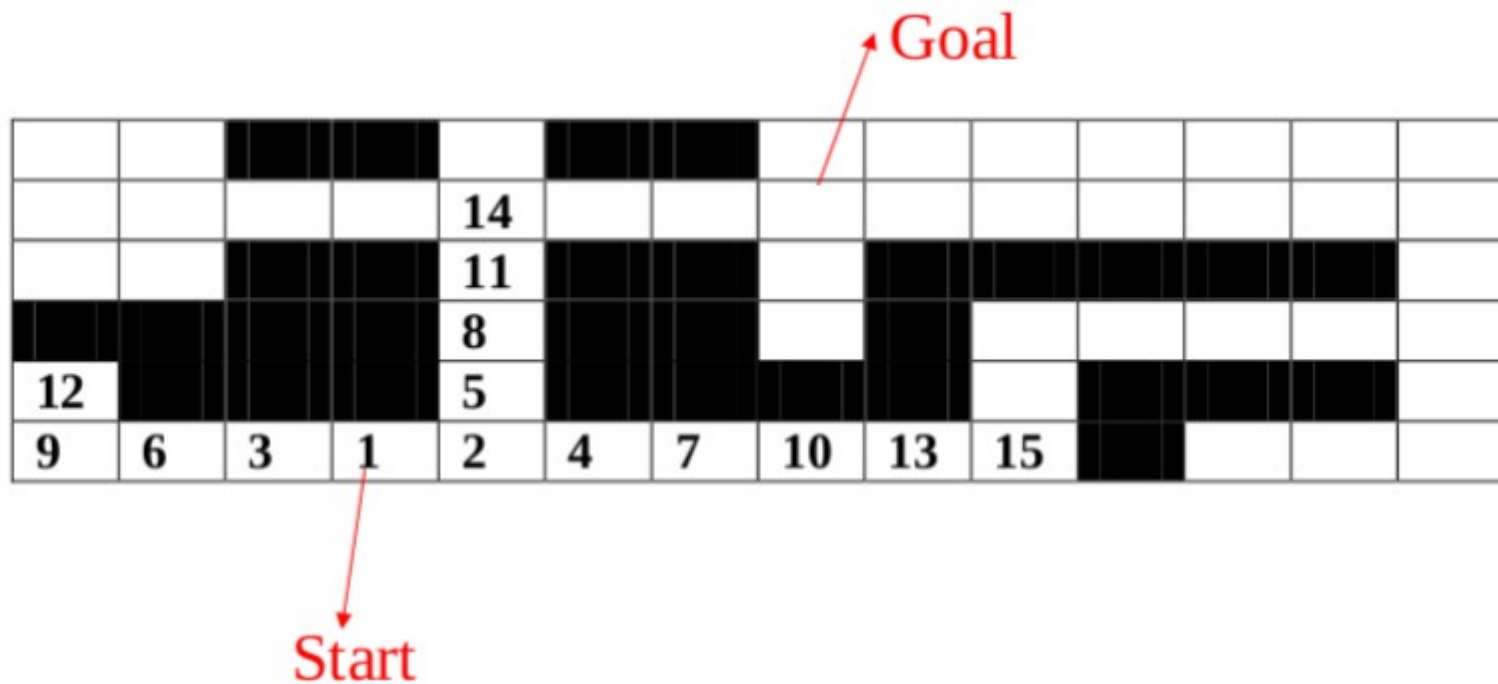
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



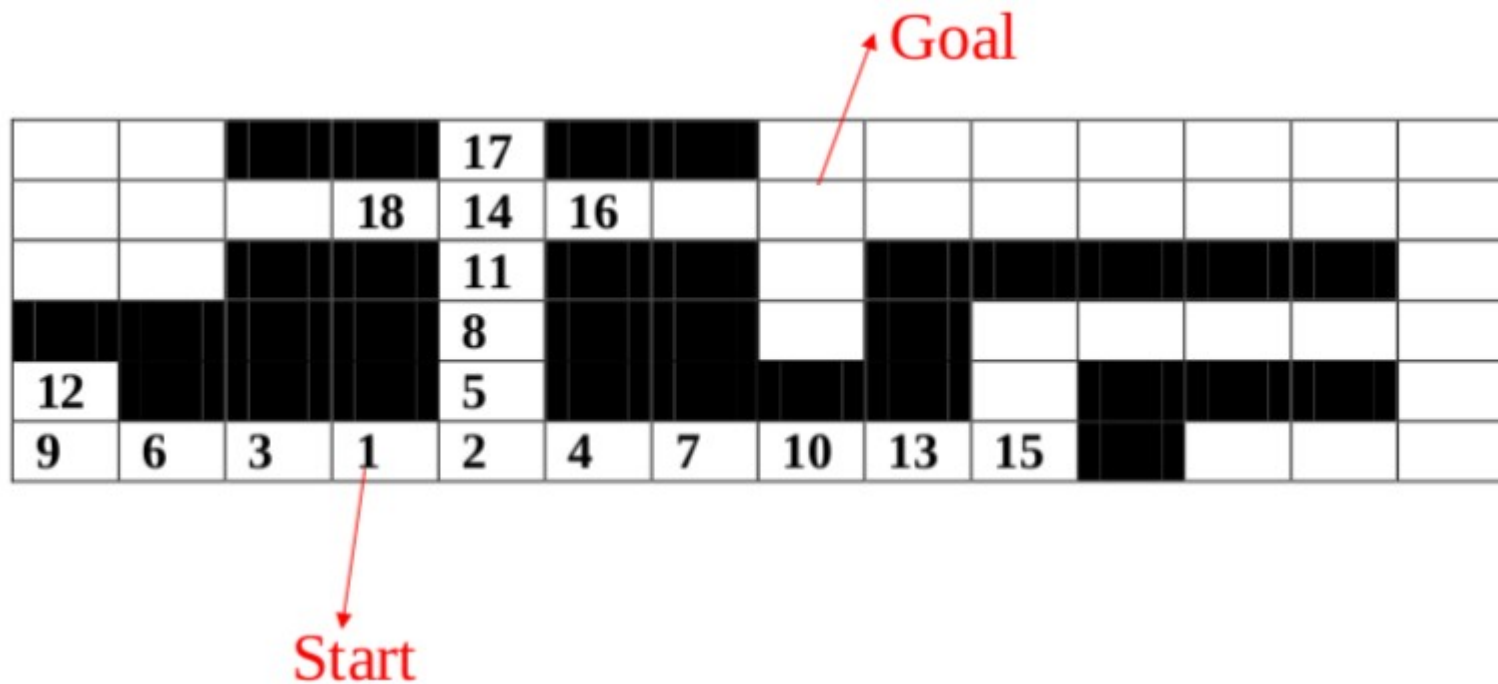
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



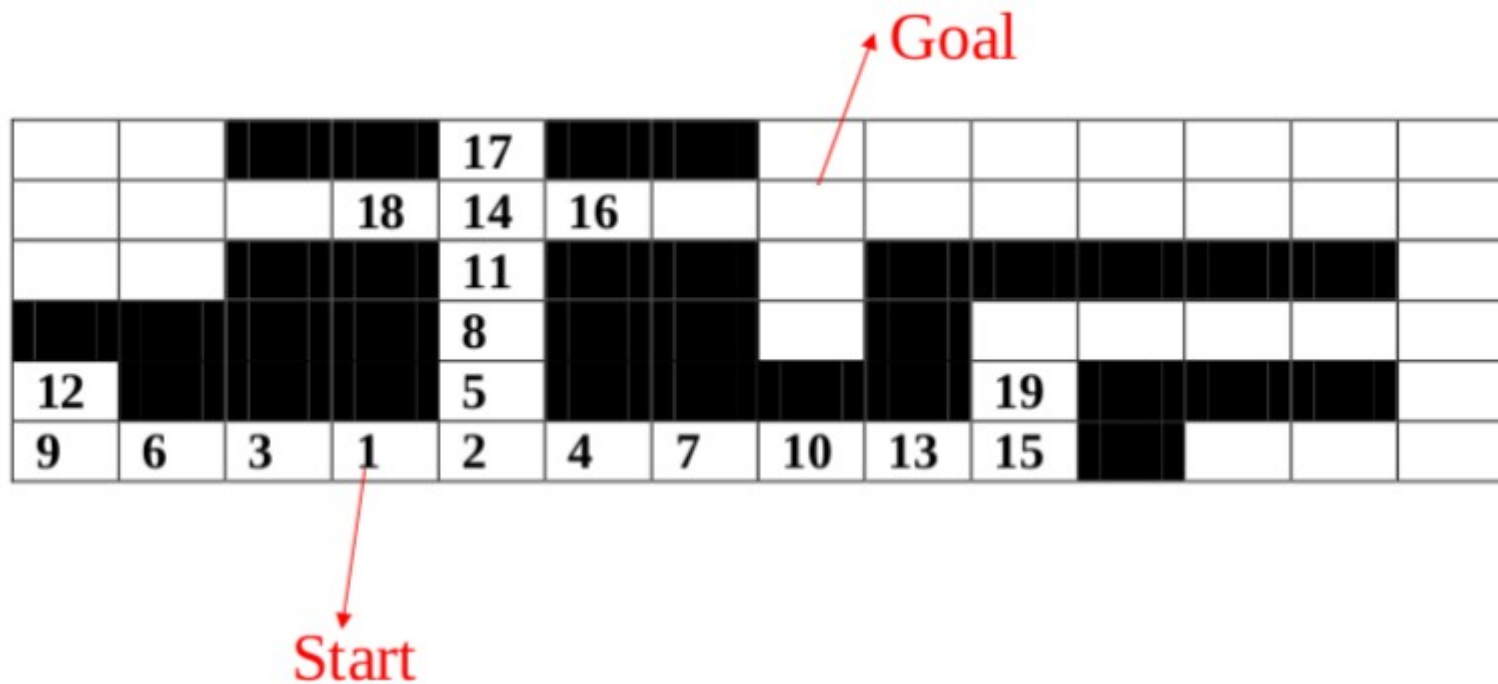
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



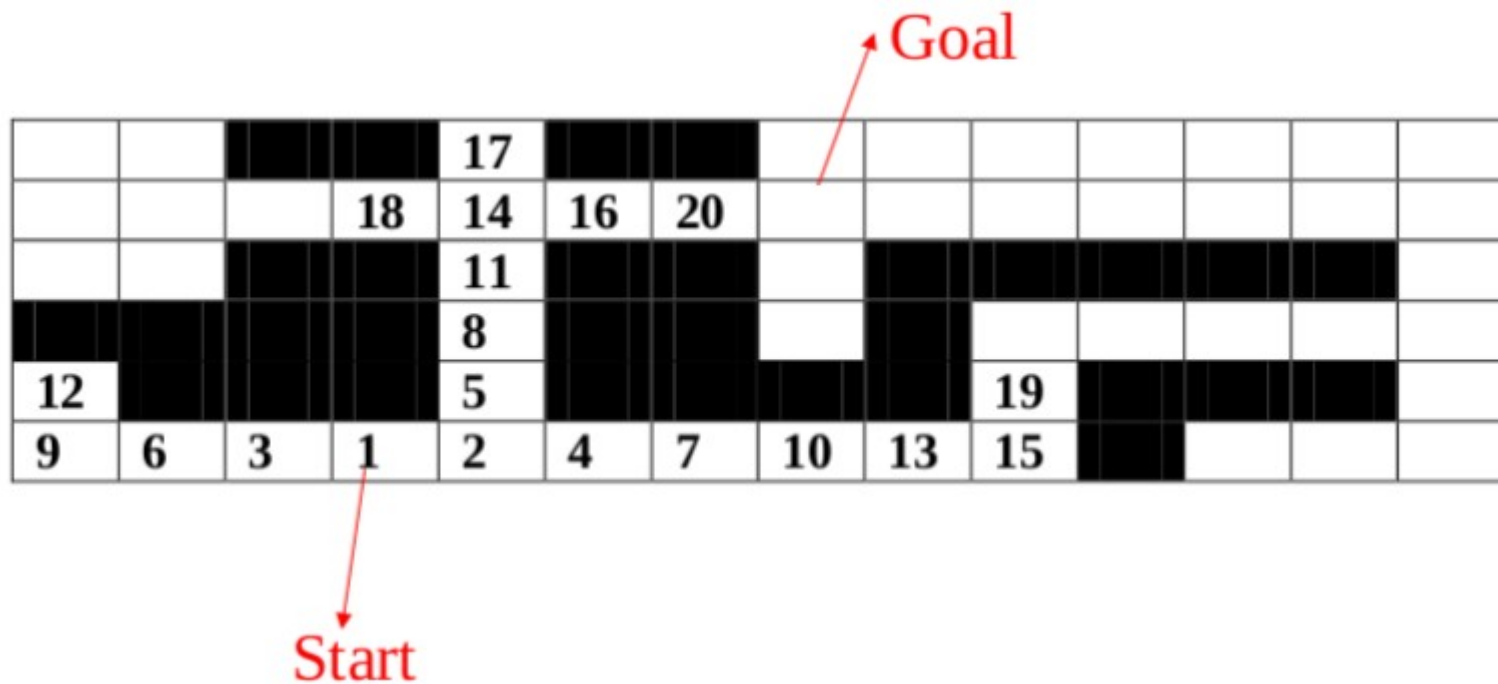
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



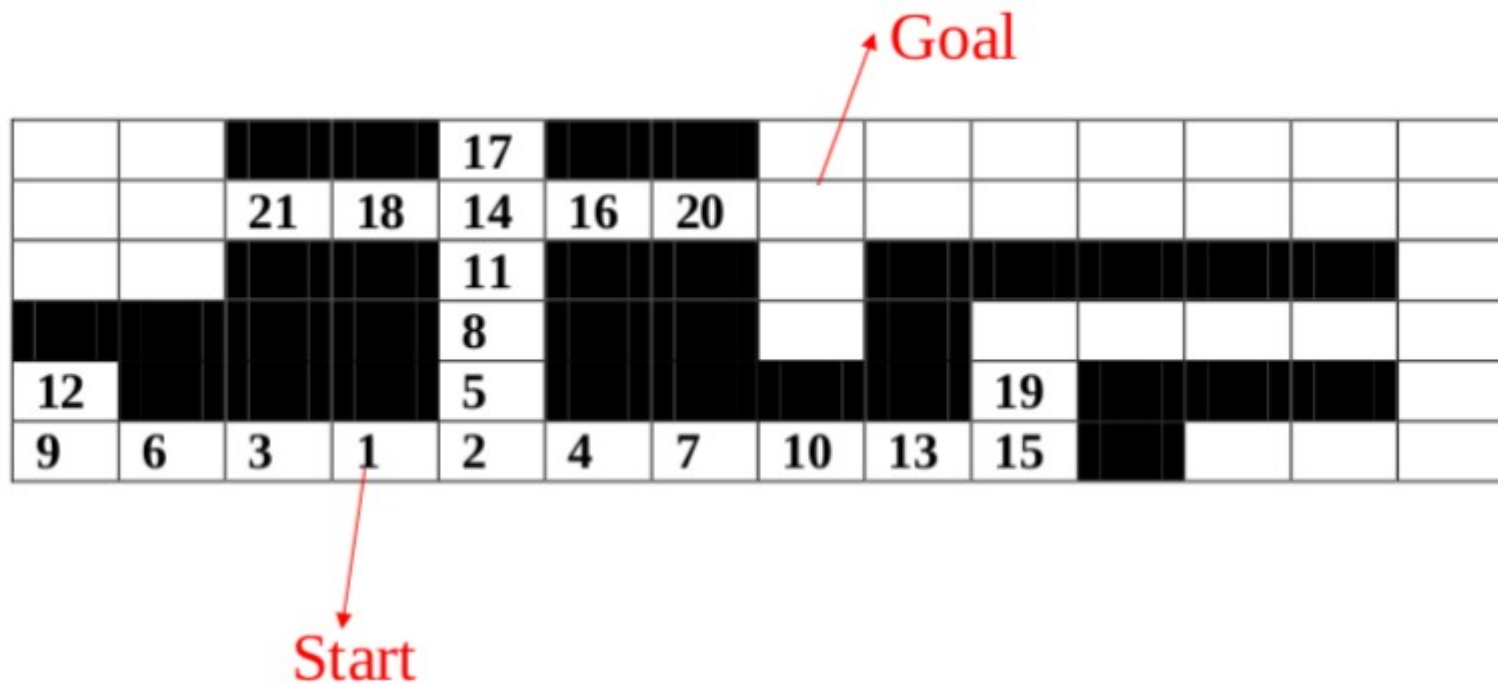
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



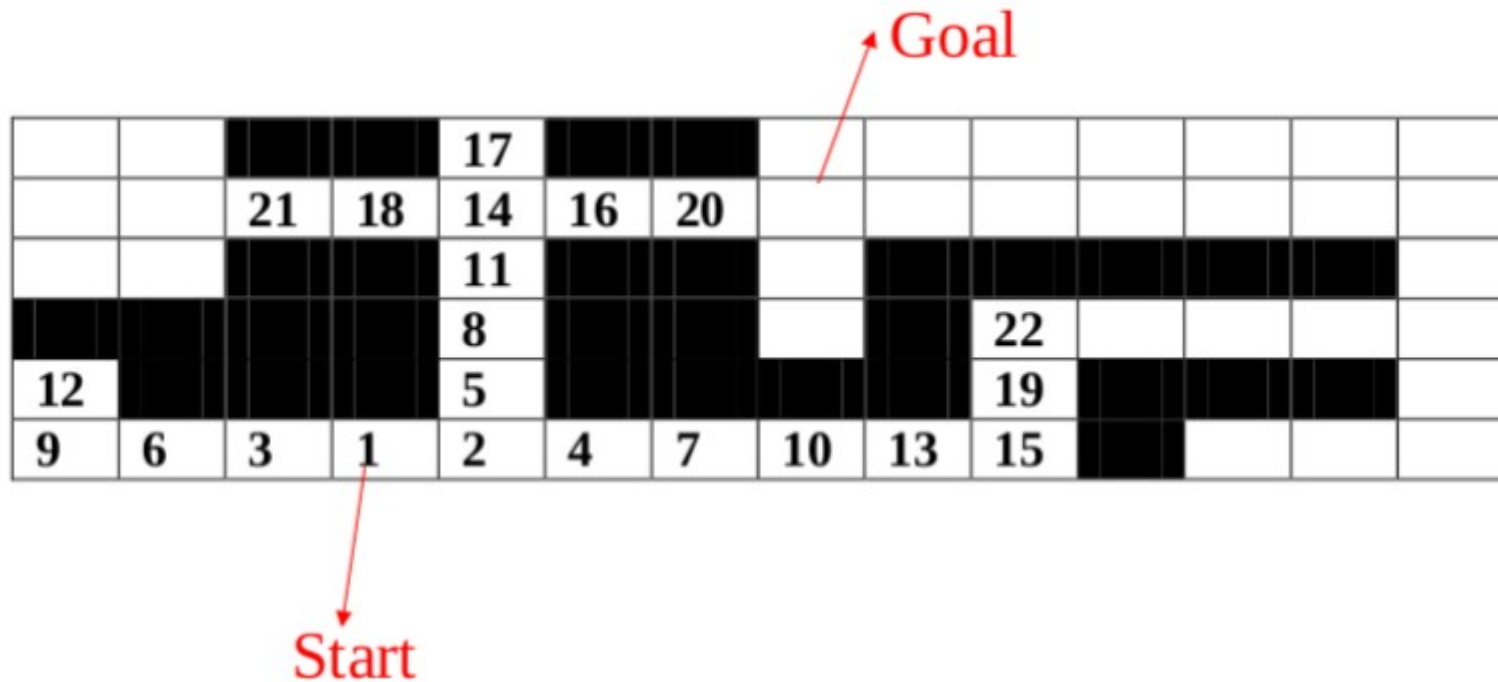
# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?





# Breadth First Search ( BFS )

- ¿ Cómo se desarrolla BFS sobre el mapa ?



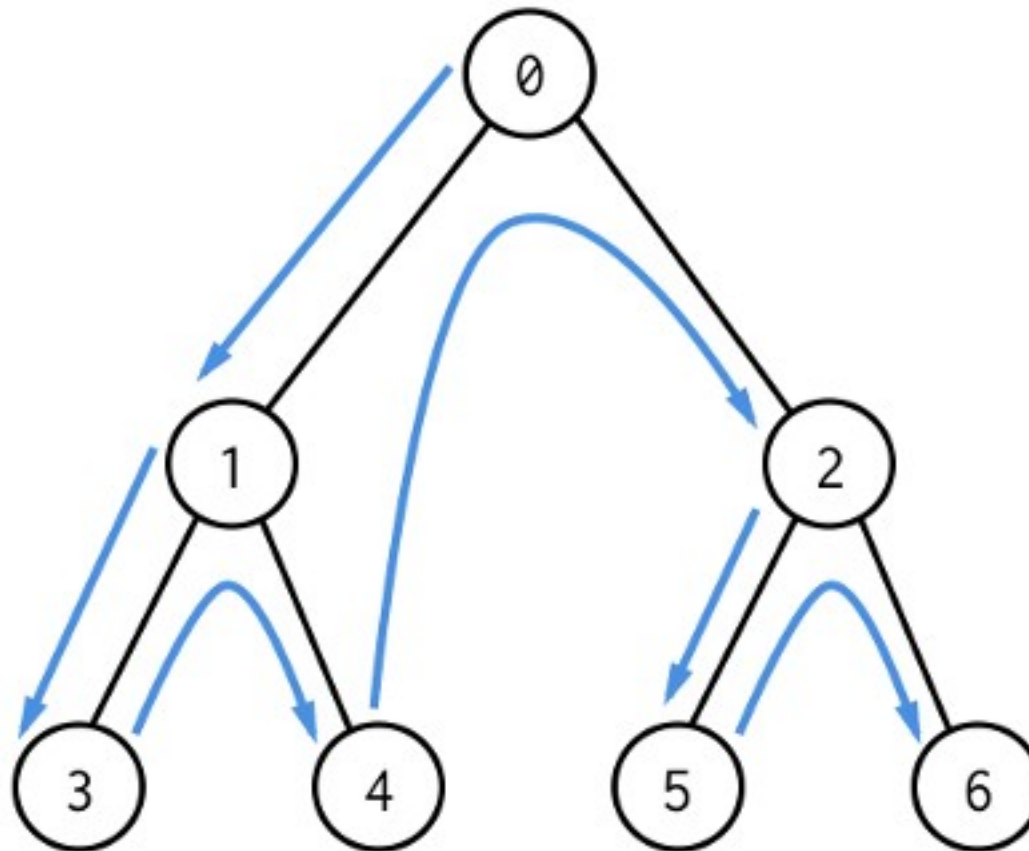
# Breadth First Search ( BFS )

## Algoritmo

```
1: procedure BFS(start_state, goal_state)
2:    $Q \leftarrow \text{Queue}()$ 
3:    $Q.\text{push\_back}(\text{start\_state})$ 
4:   while  $Q$  not empty do
5:      $s = Q.\text{pop\_front}()$                                 ▷ get older element
6:     if  $s$  is equal  $\text{goal\_state}$  then
7:        $\text{return } s$ 
8:     for all  $\text{succesors}$  in  $s.\text{expand}()$  do                ▷ successors linked to its parent
9:       if  $\text{successor}$  not discovered then
10:         $\text{successor.set\_discovered}()$ 
11:         $Q.\text{push\_back}(\text{successor})$ 
```

# Depth First Search ( DFS )

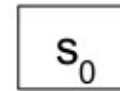
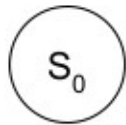
- Búsqueda en profundidad
- Orden de expansión vertical y luego horizontal



# Depth First Search ( DFS )

**Ejemplo:** Start =  $s_0$ , Goal =  $s_8$

- Se expande estado  $s_0$  y se agrega a una cola de estados

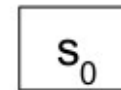
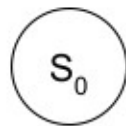


- Se saca el estado *más nuevo* desde la cola, y se expande

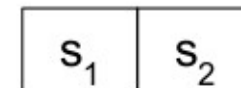
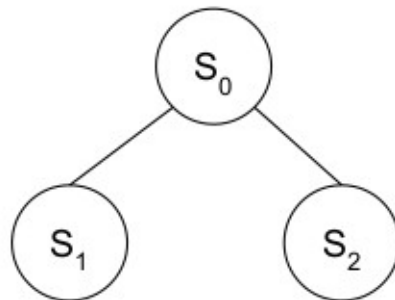
# Depth First Search ( DFS )

**Ejemplo:** Start =  $s_0$ , Goal =  $s_8$

- Se expande estado  $s_0$  y se agrega a una cola de estados

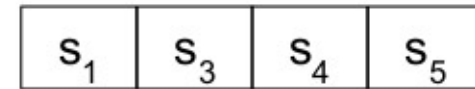
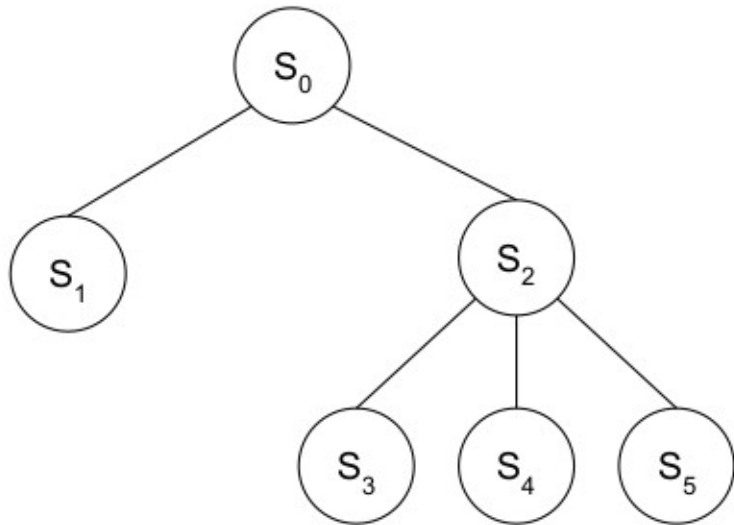


- Se saca el estado *más nuevo* desde la cola, y se expande
- Estados hijos se almacenan en la cola



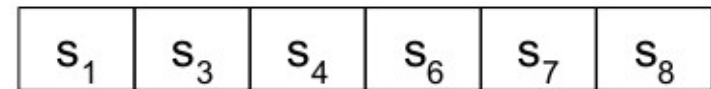
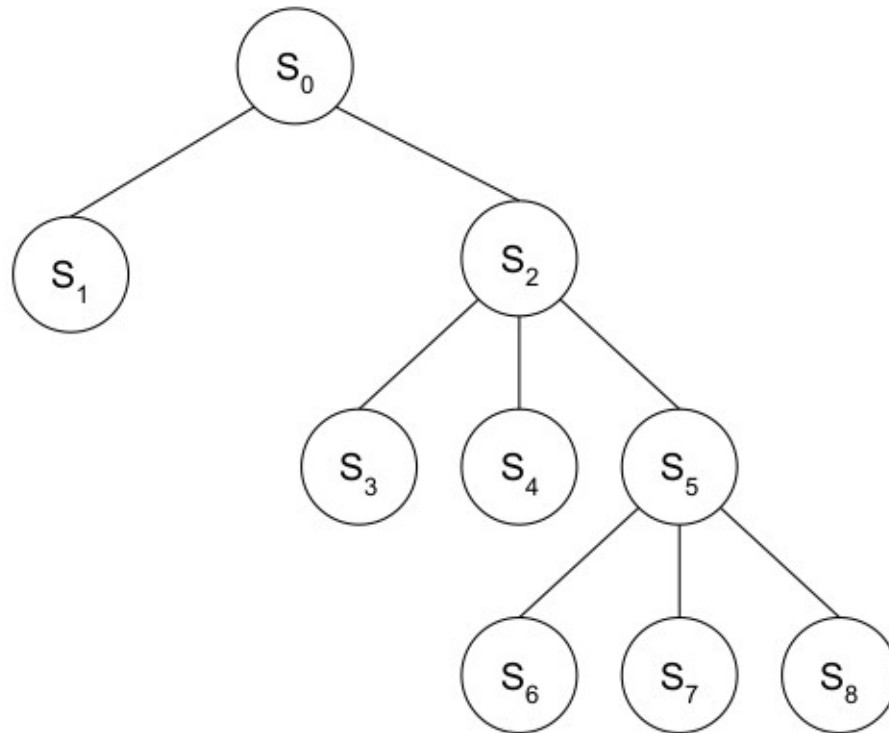
# Depth First Search ( DFS )

- Se saca el estado *más nuevo* desde la cola, y se expande
- Estados hijos se almacenan en la cola



# Depth First Search ( DFS )

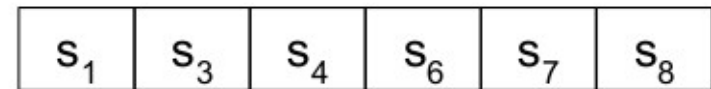
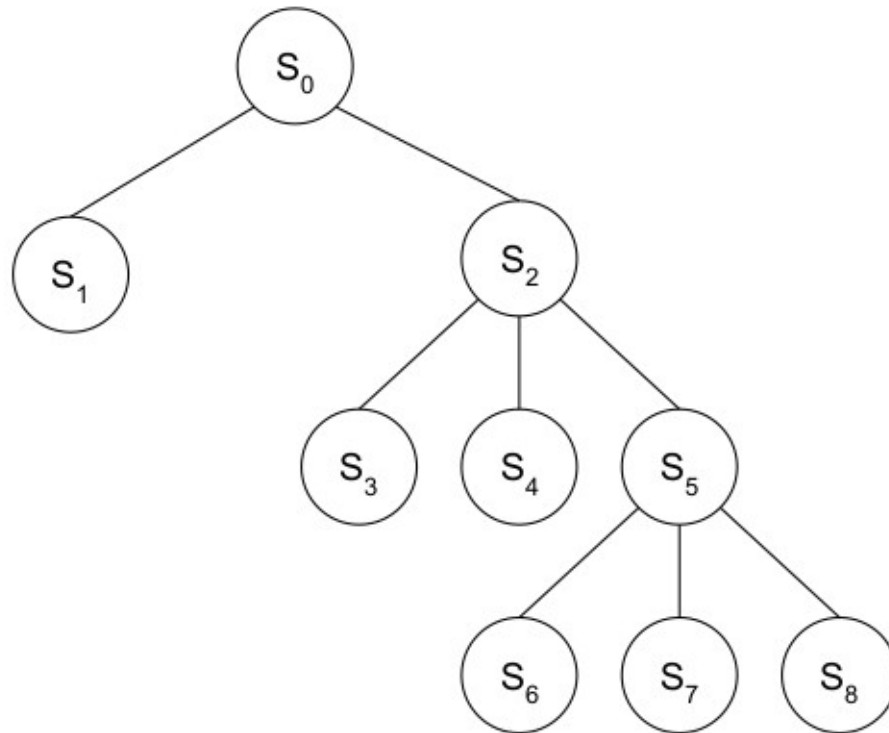
- Se saca el estado *más nuevo* desde la cola, y se expande
- Estados hijos se almacenan en la cola



- Cola de expansión es del tipo ...

# Depth First Search ( DFS )

- Se saca el estado *más antiguo* desde la cola, y se expande
- Estados hijos se almacenan en la cola

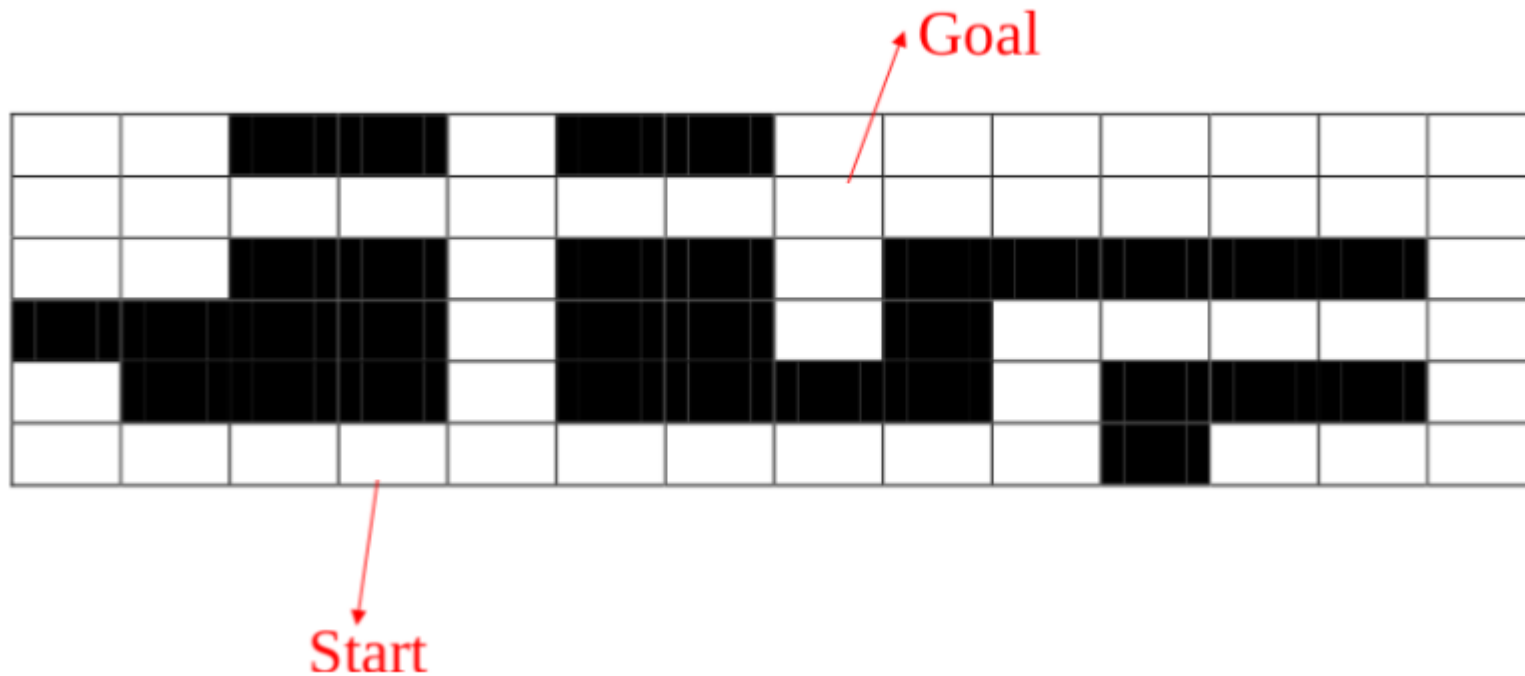


- Cola de expansión es del tipo **LIFO**



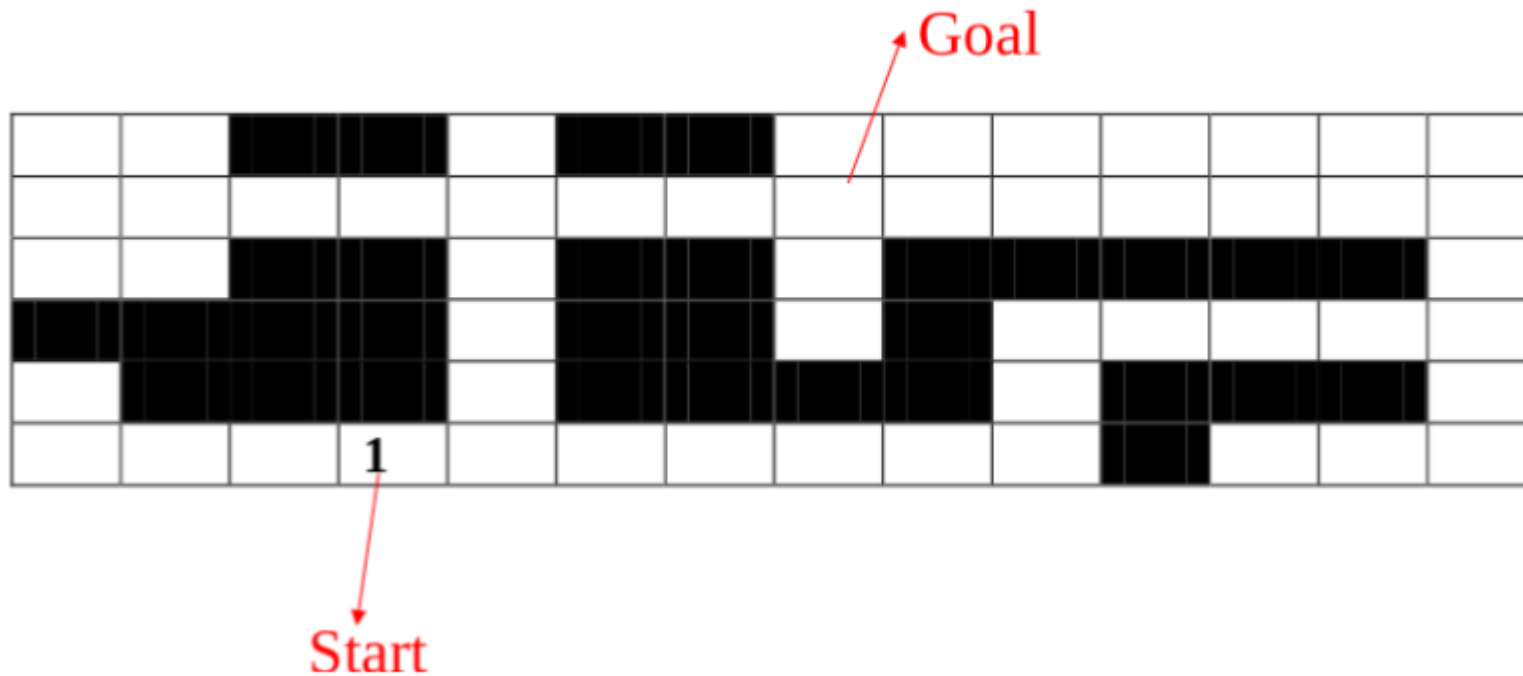
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



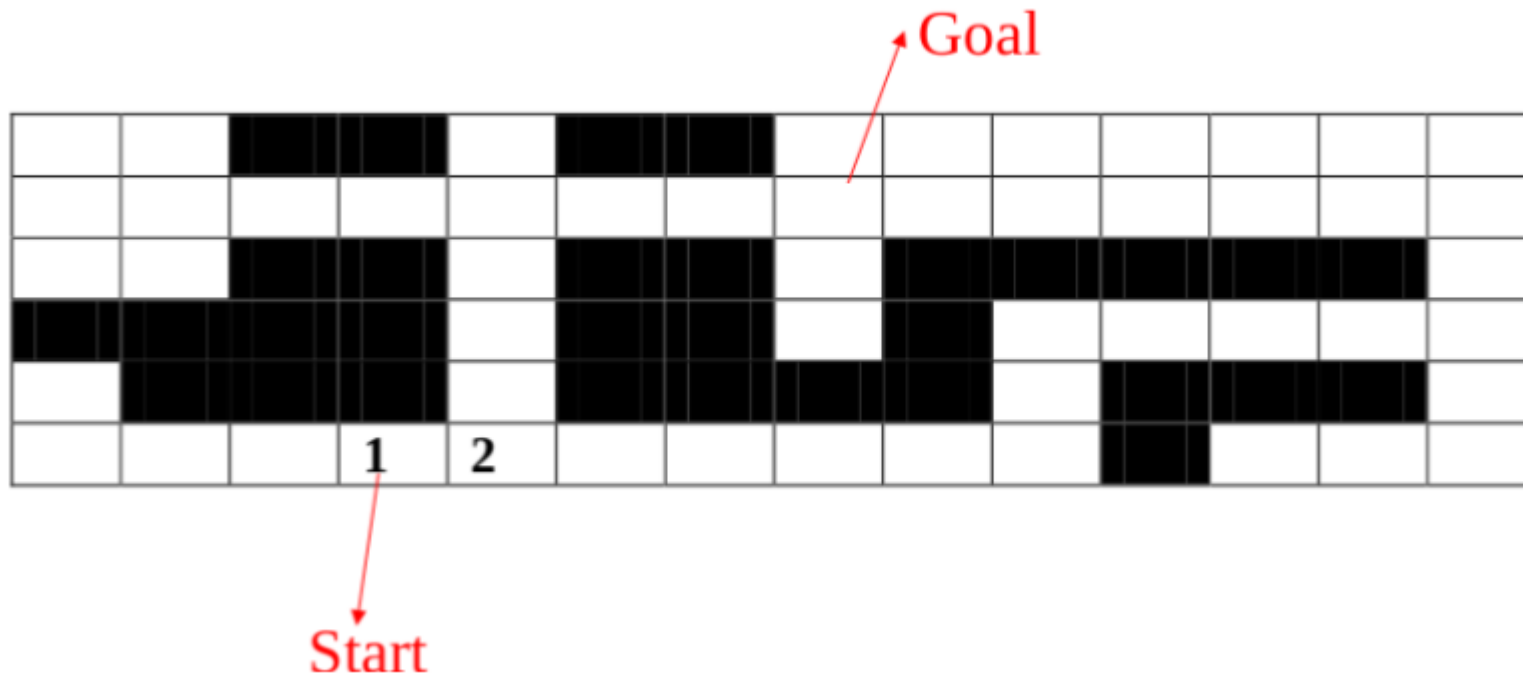
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



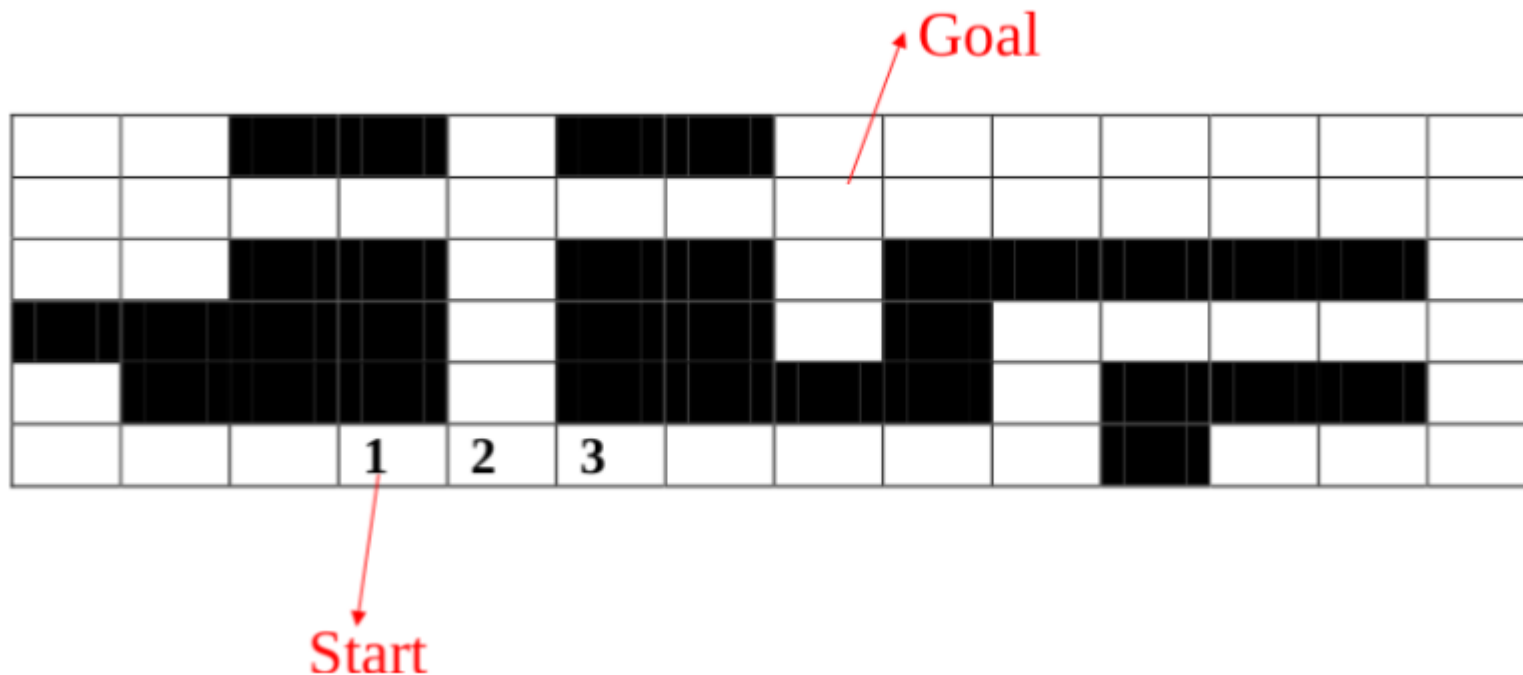
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



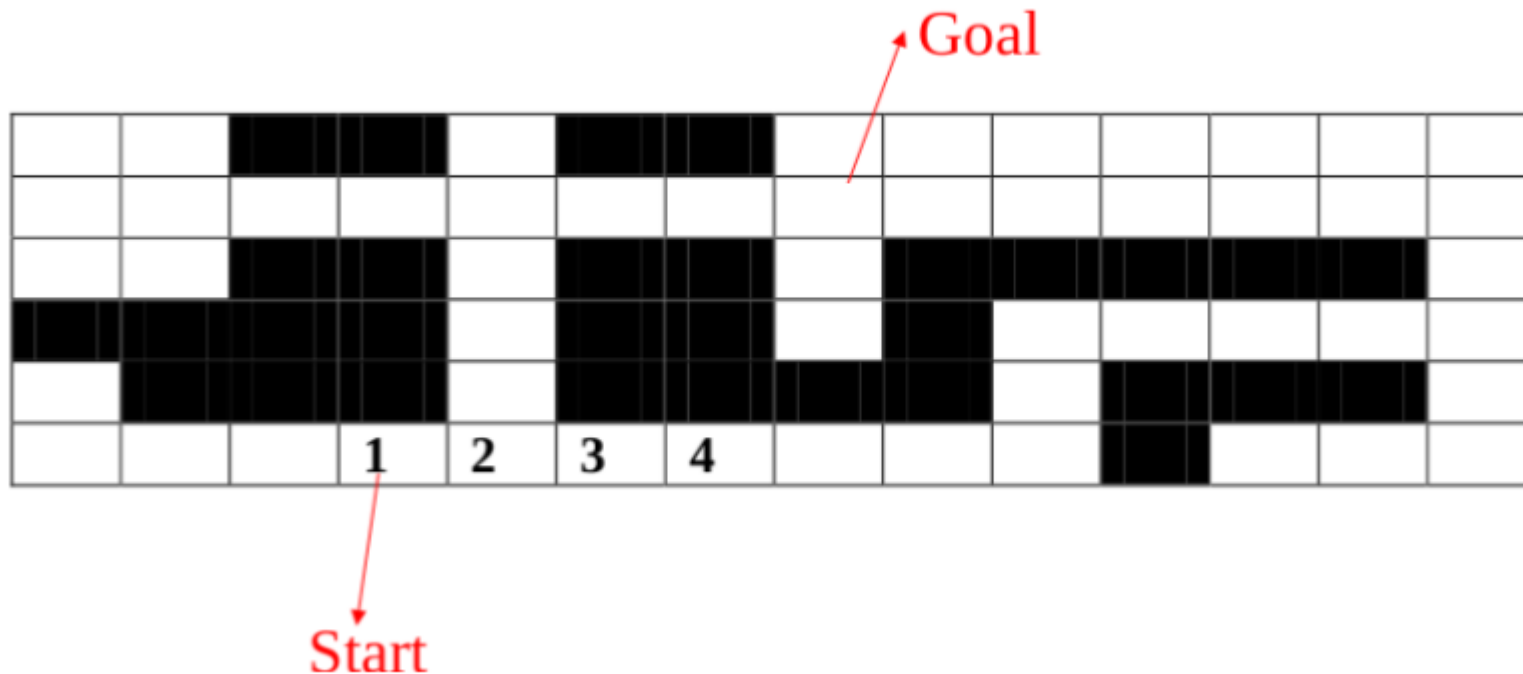
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



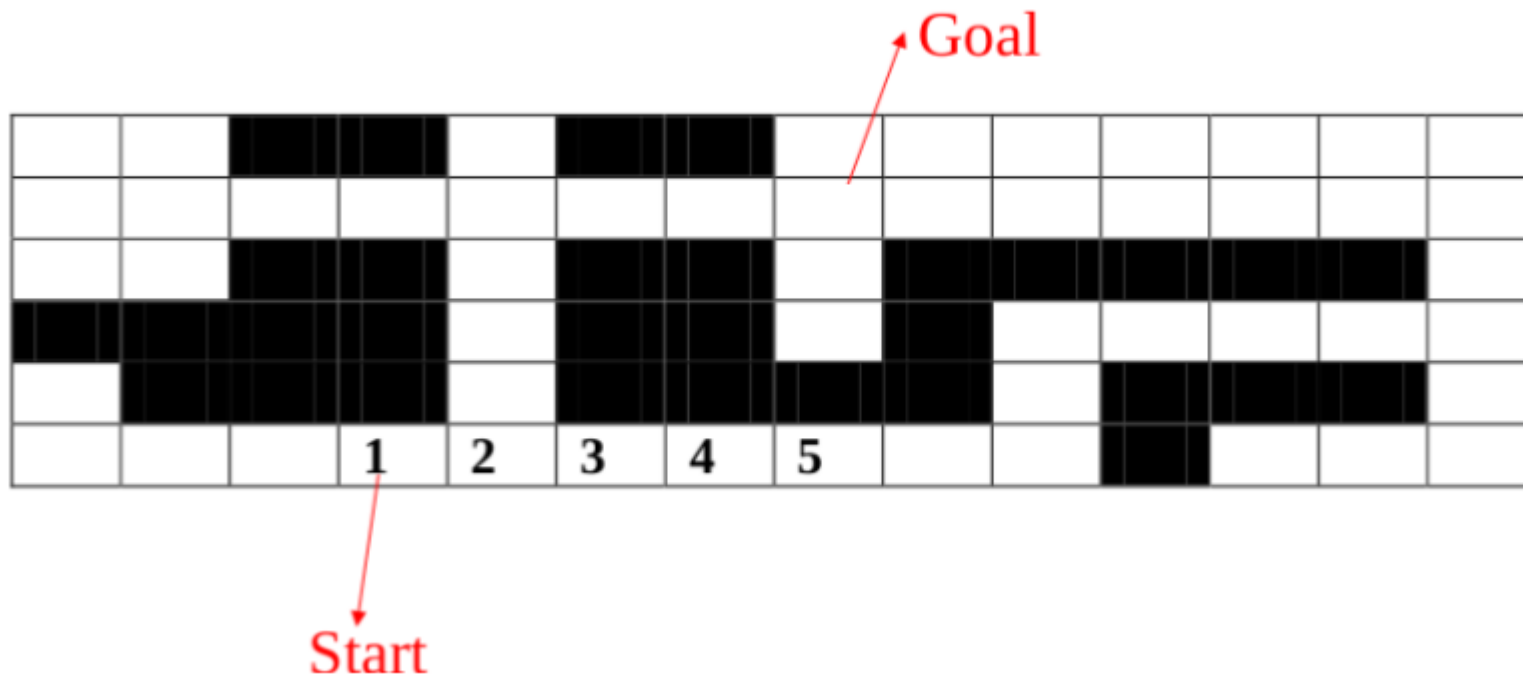
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



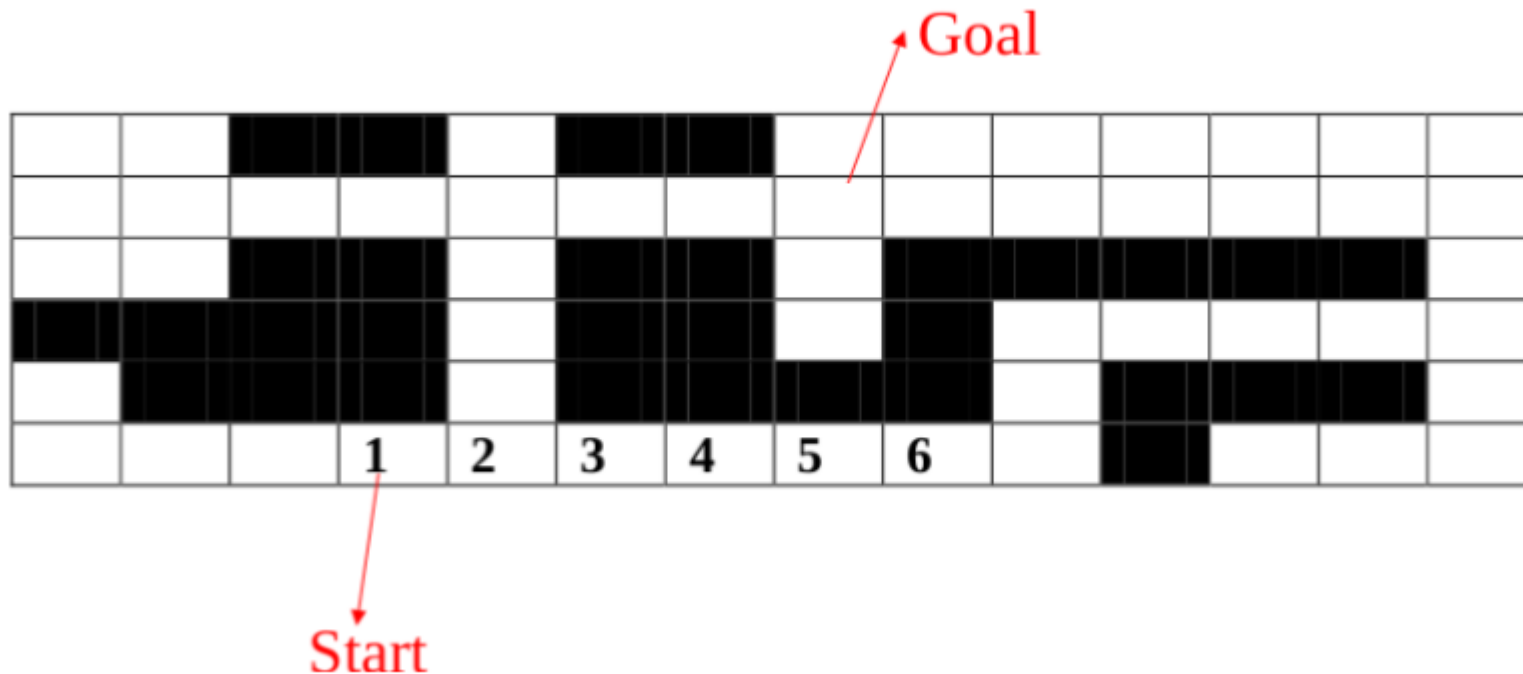
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



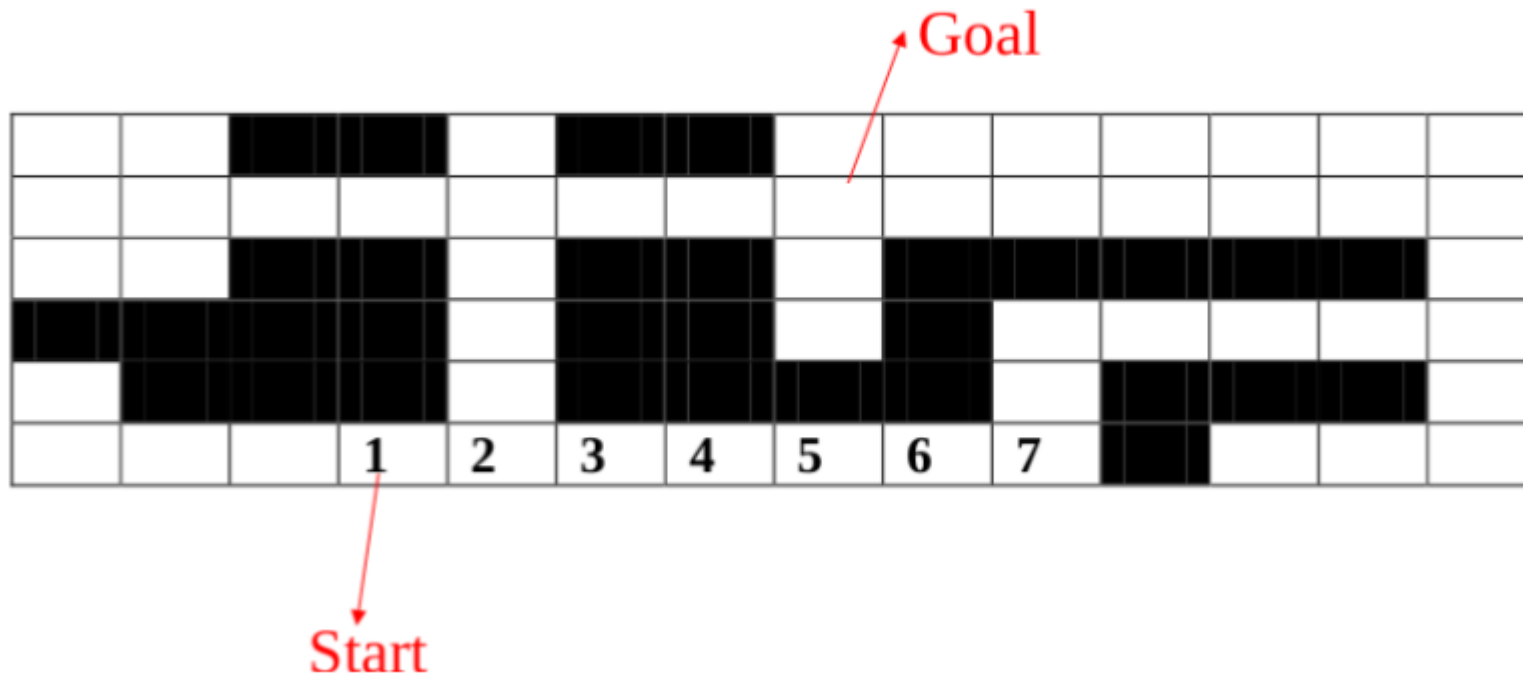
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



# Depth First Search ( DFS )

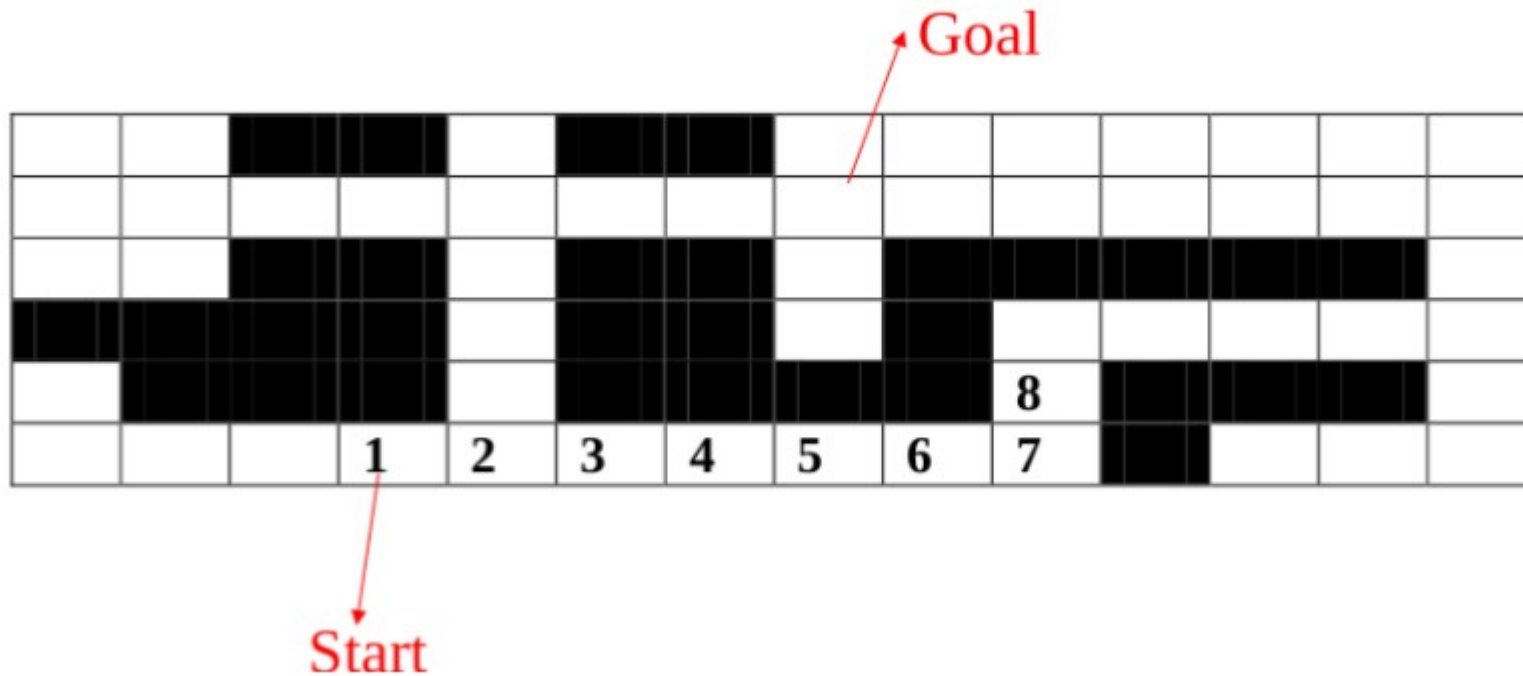
- ¿ Cómo se desarrolla DFS sobre el mapa ?





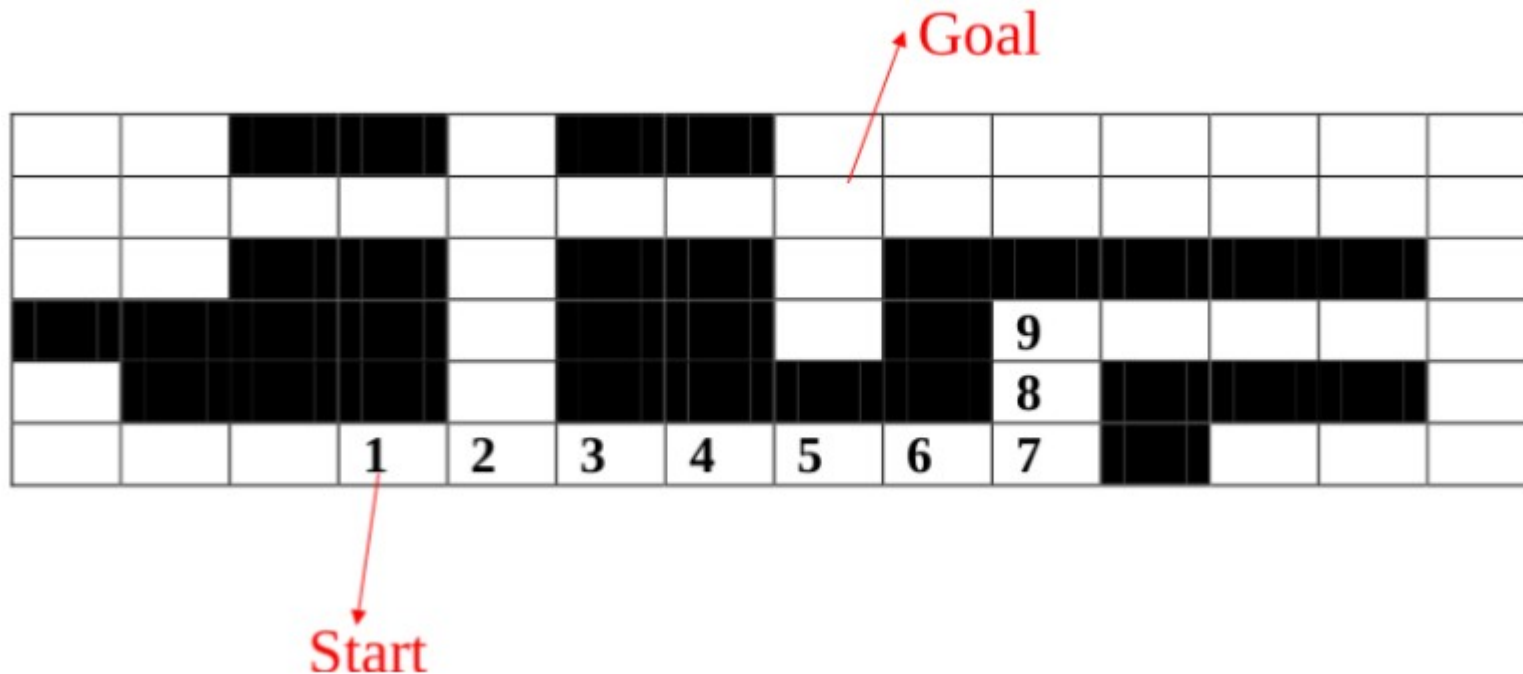
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



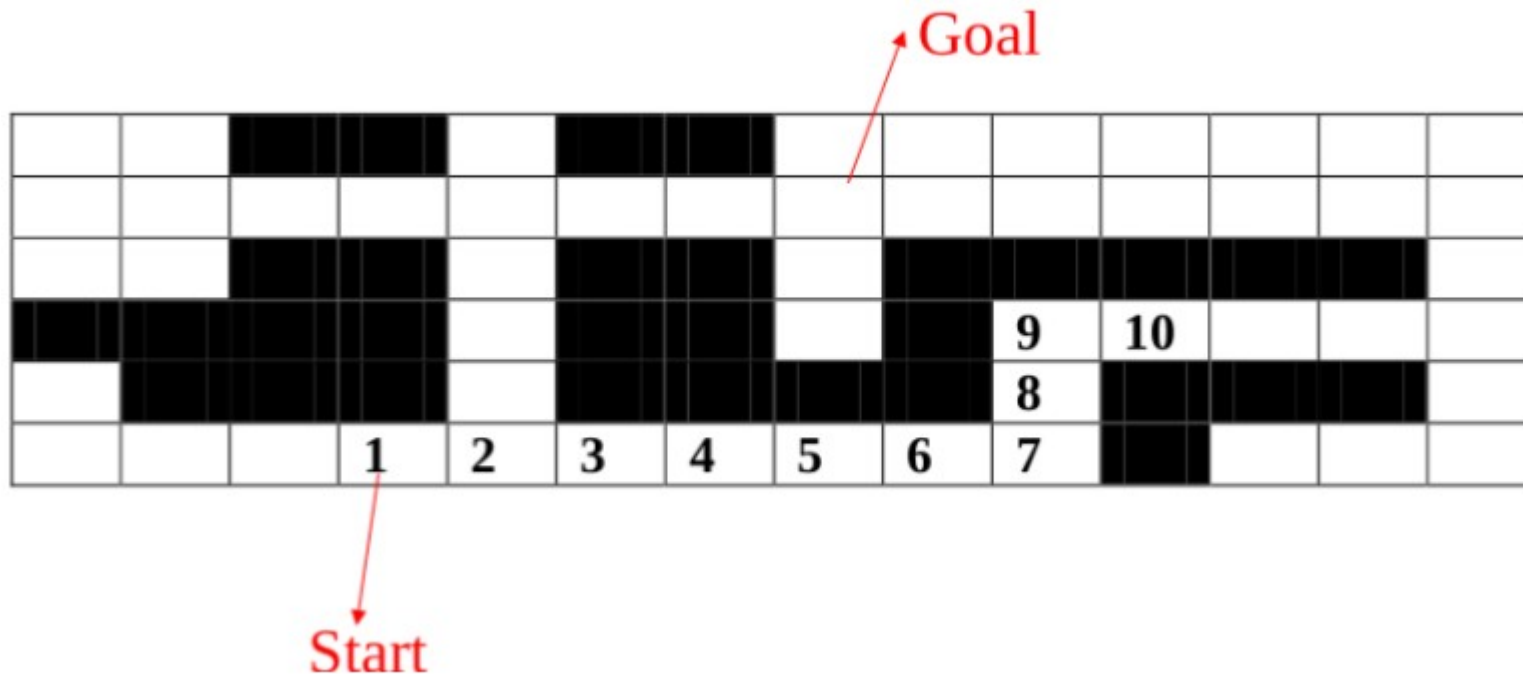
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



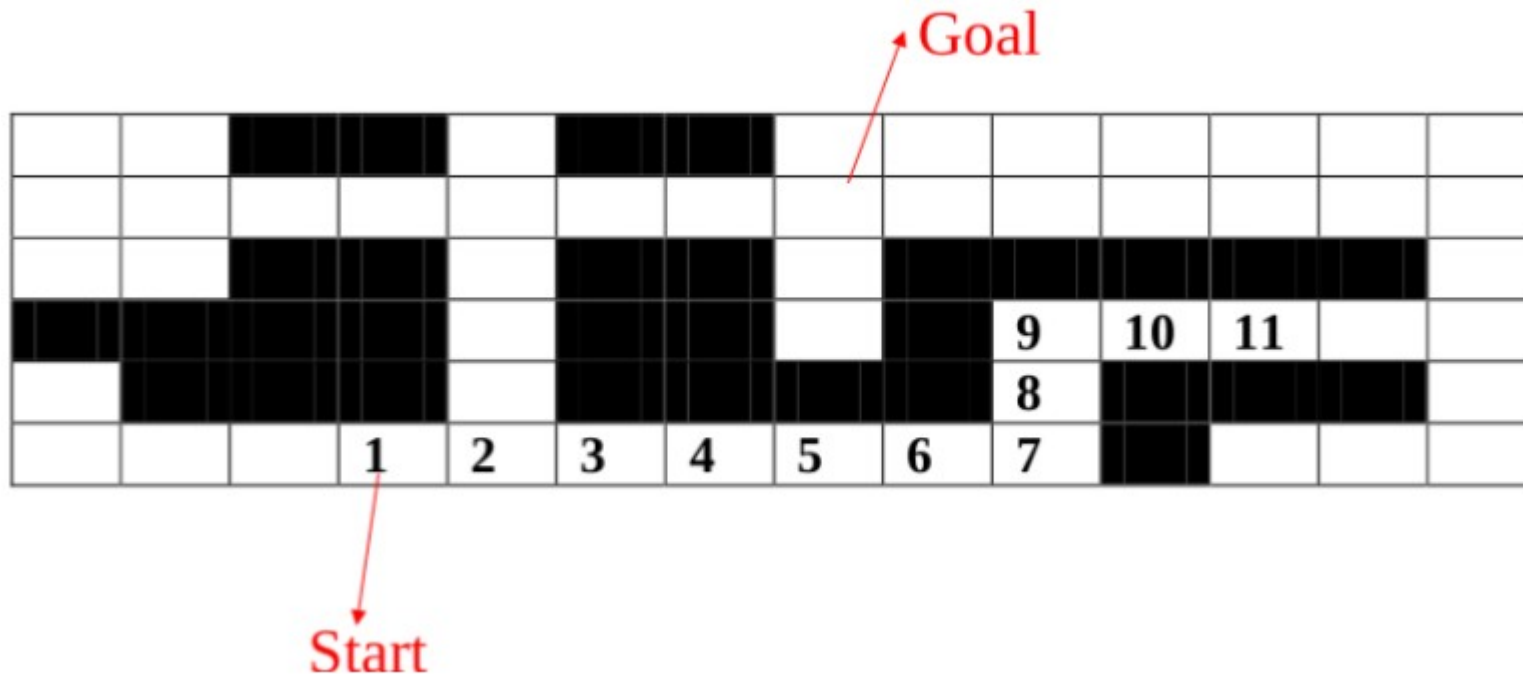
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



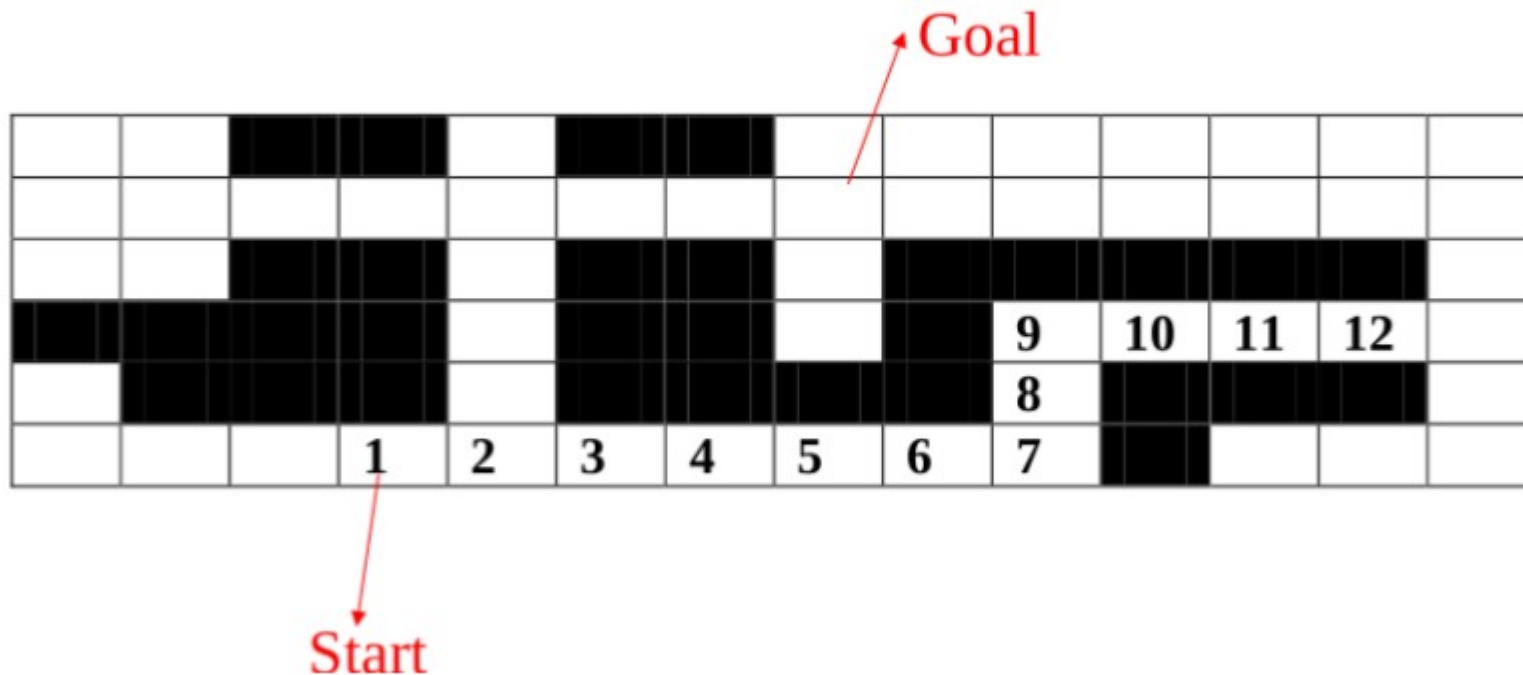
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



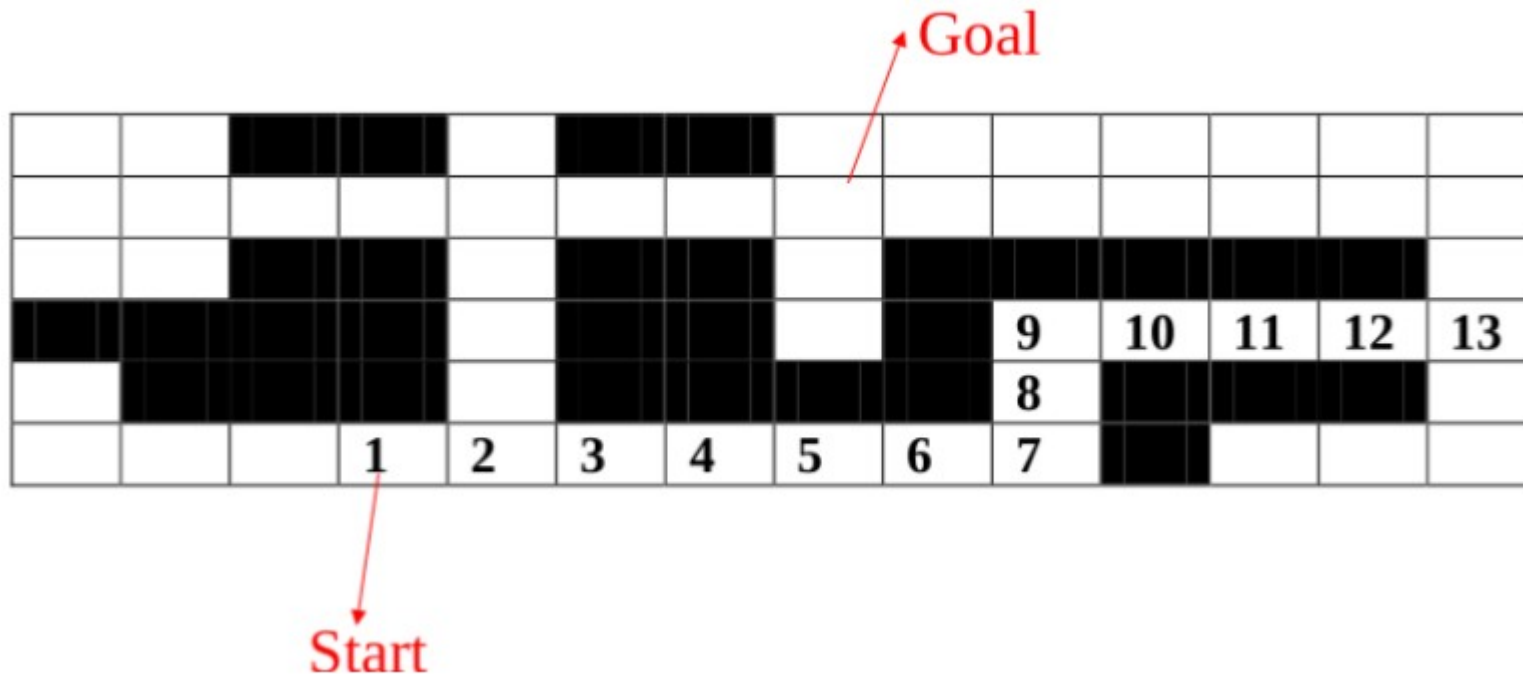
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



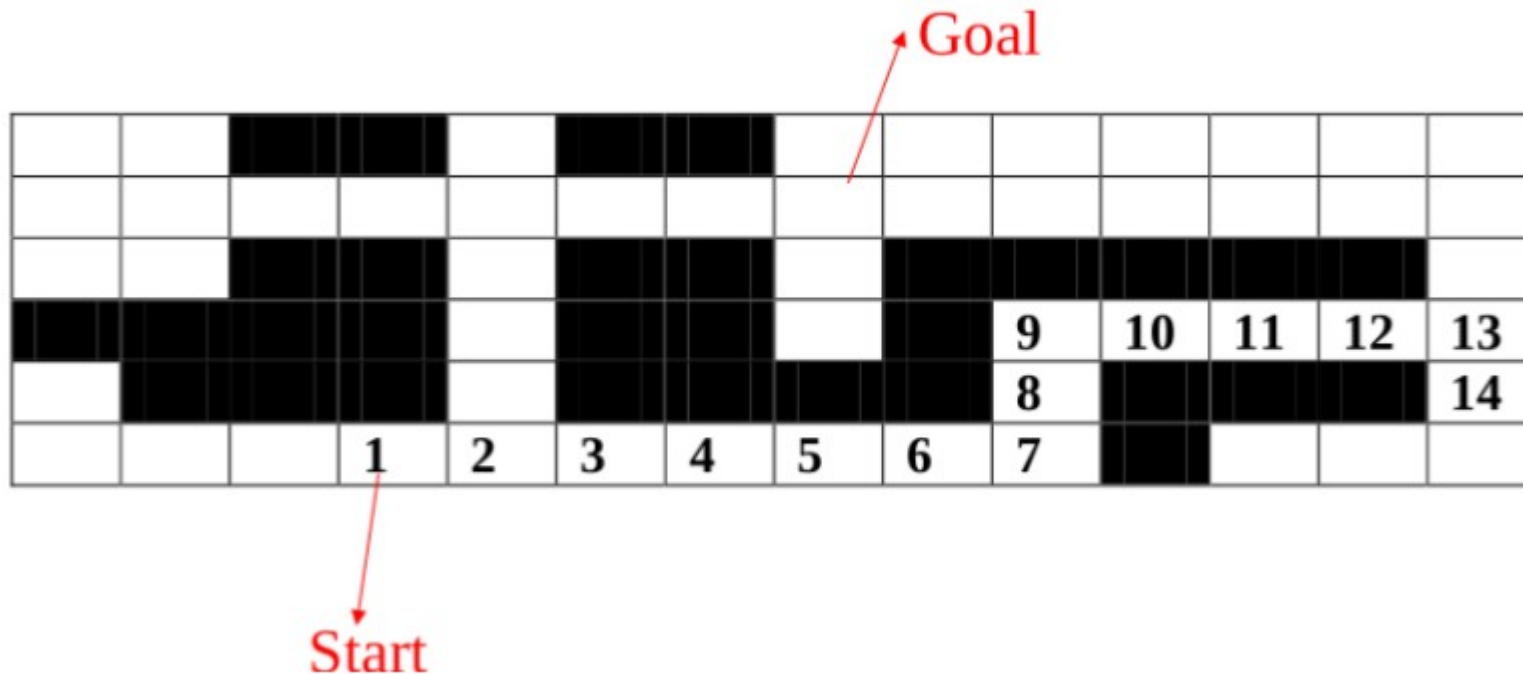
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



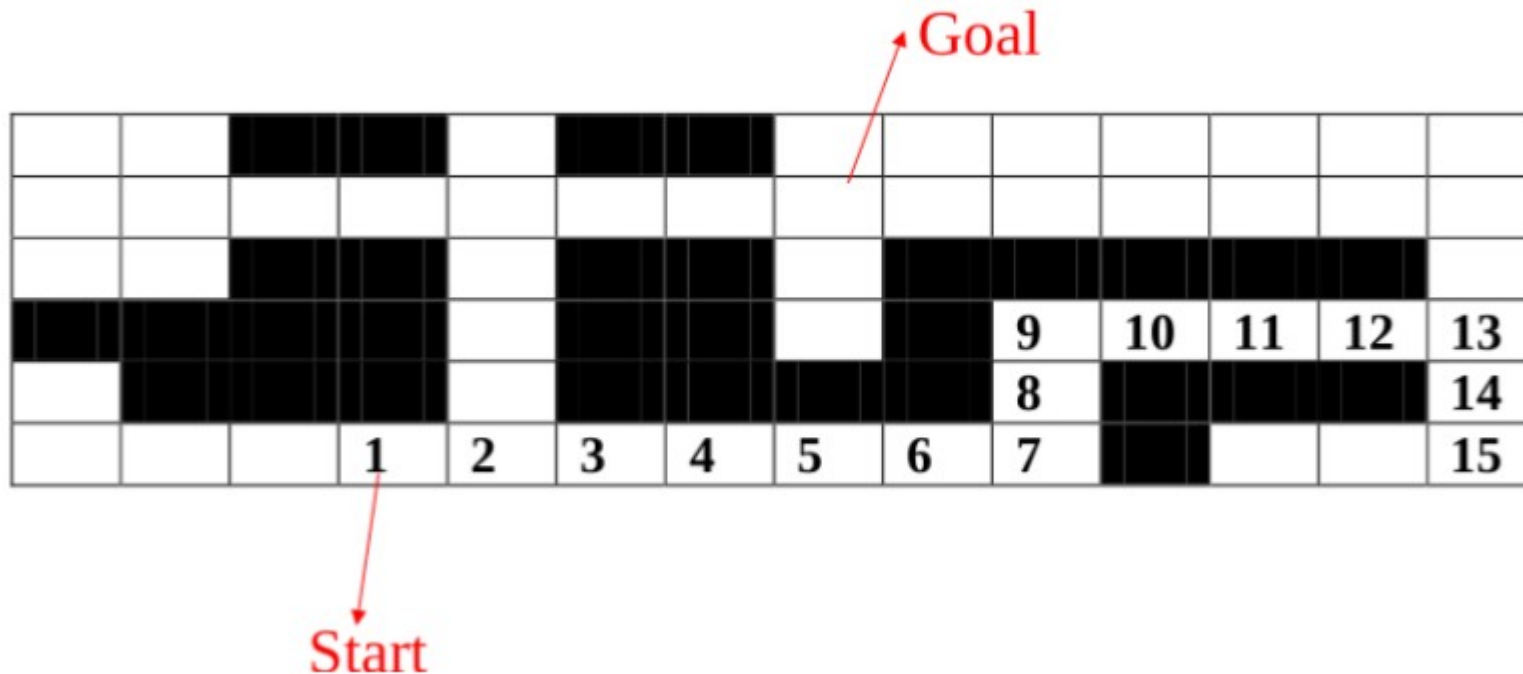
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



# Depth First Search ( DFS )

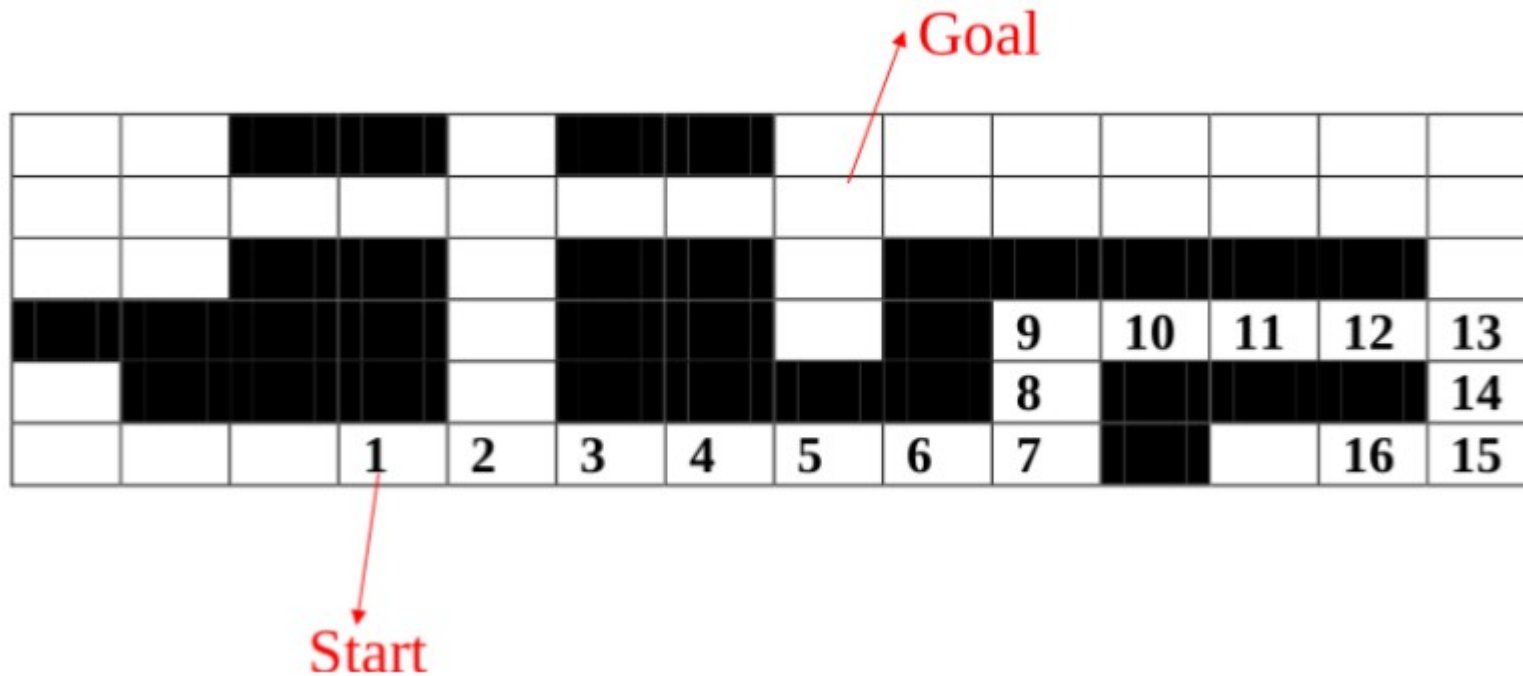
- ¿ Cómo se desarrolla DFS sobre el mapa ?





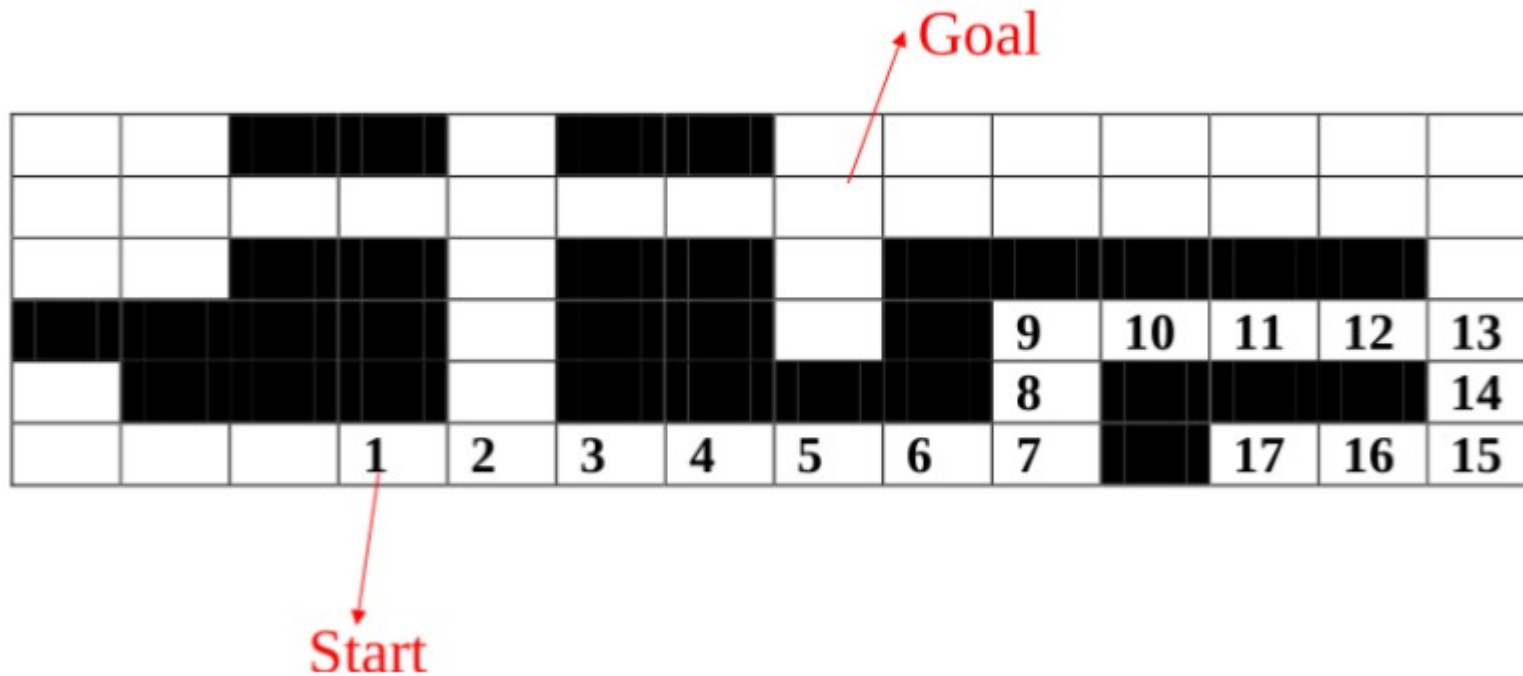
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



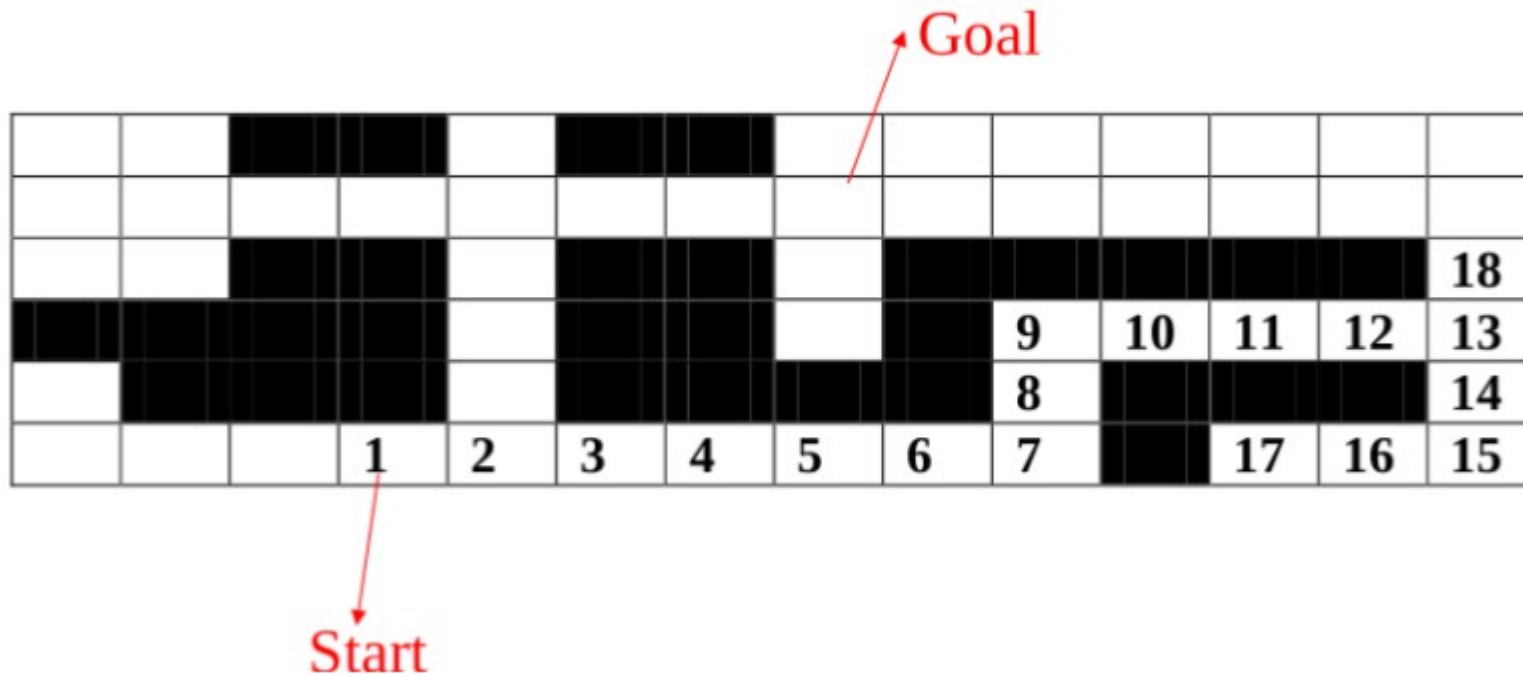
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



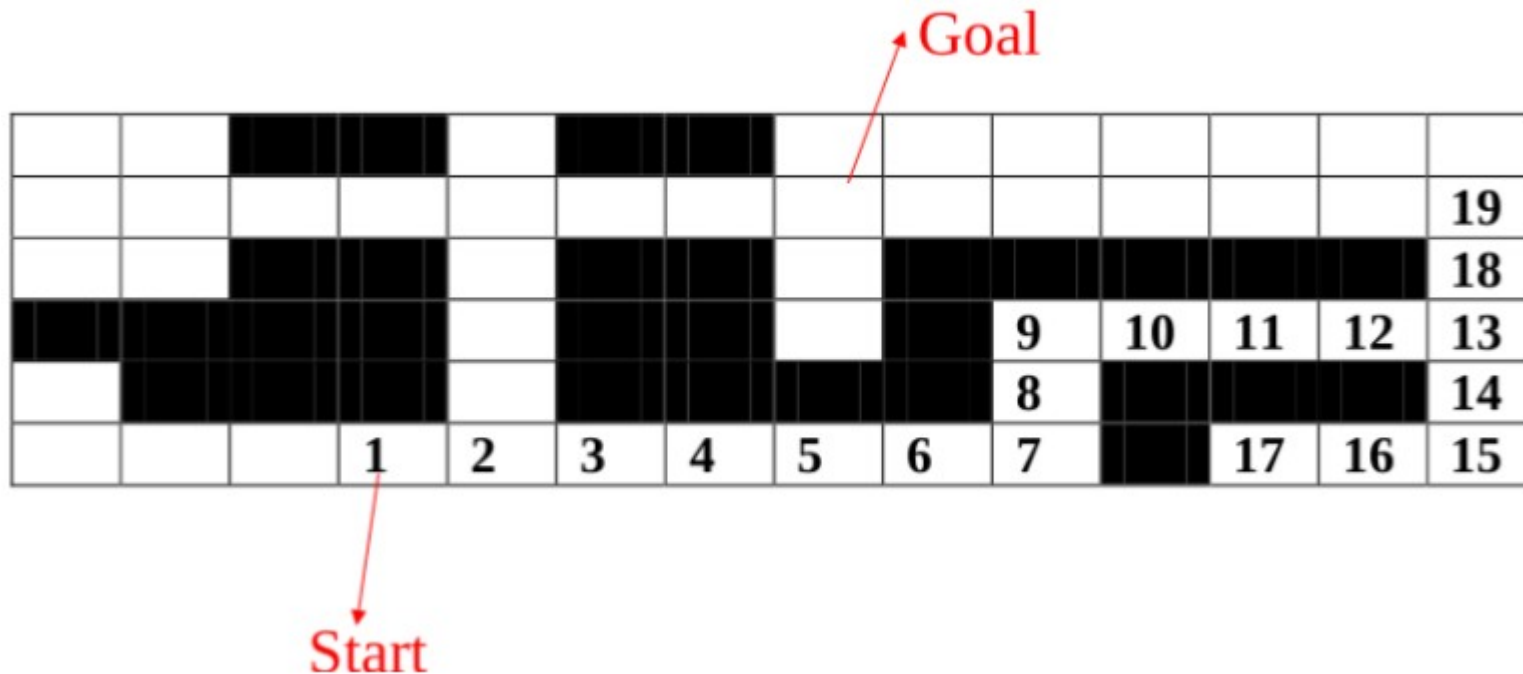
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



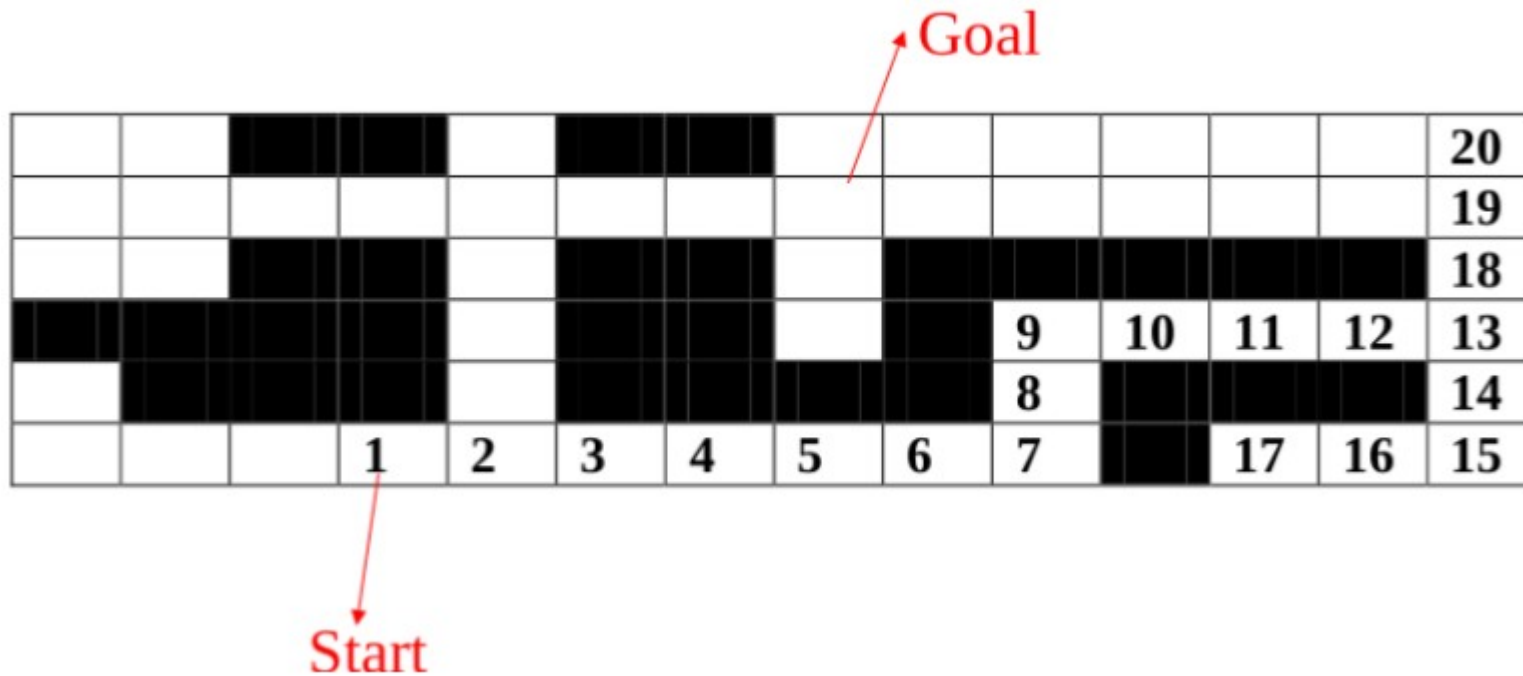
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



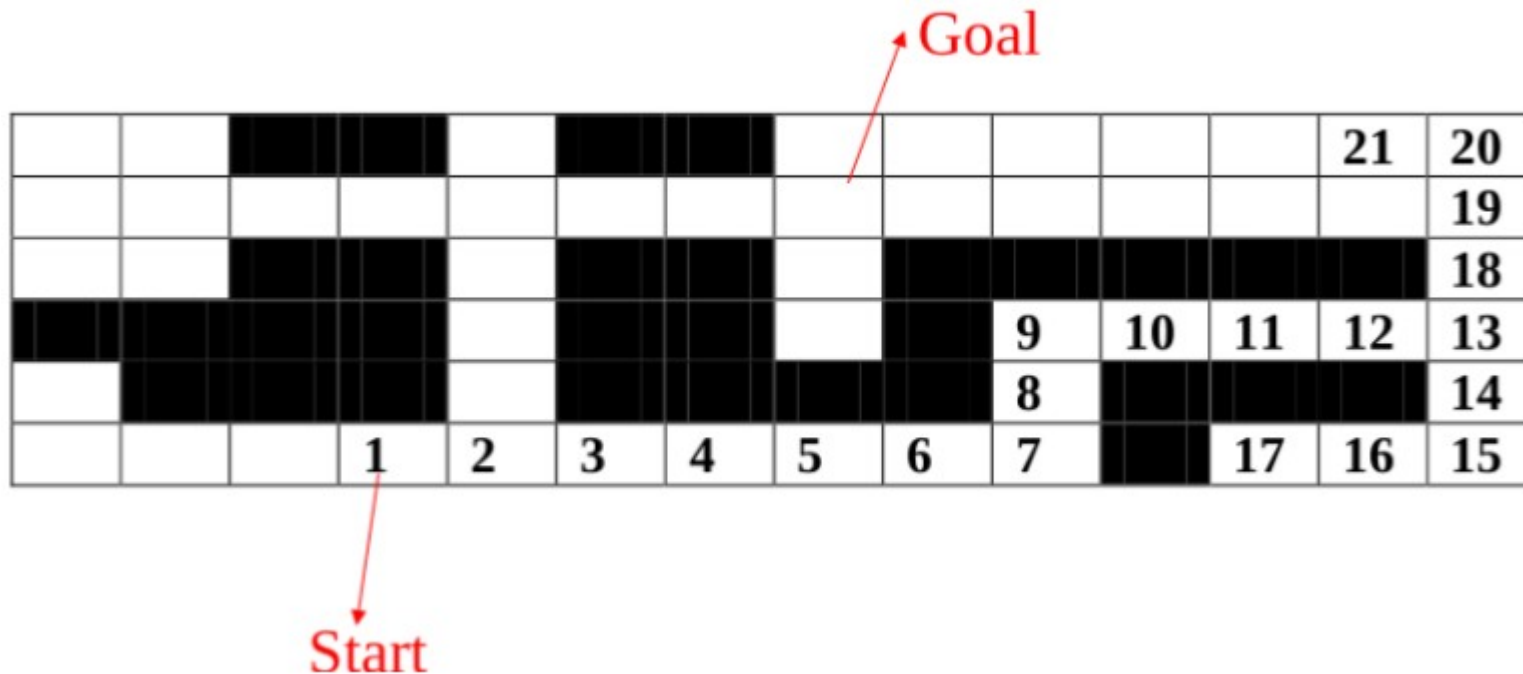
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



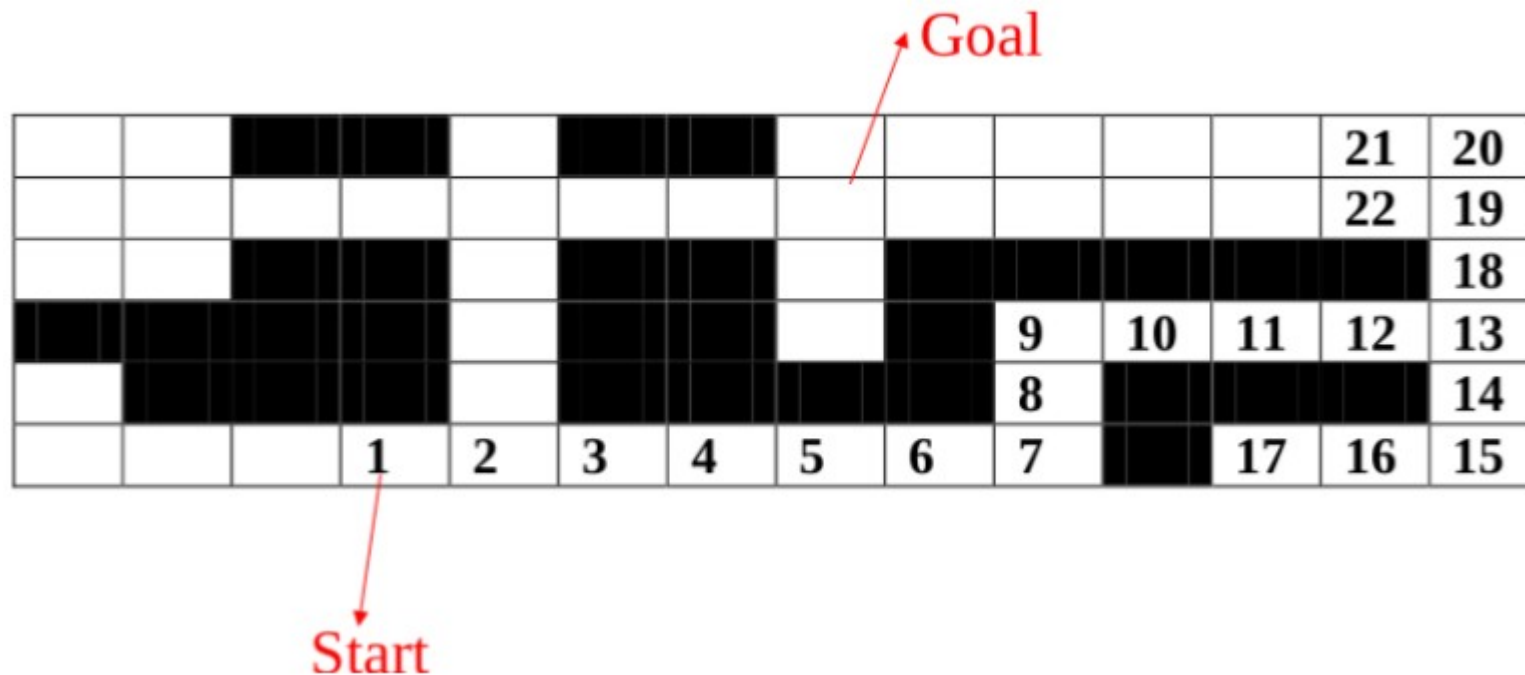
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



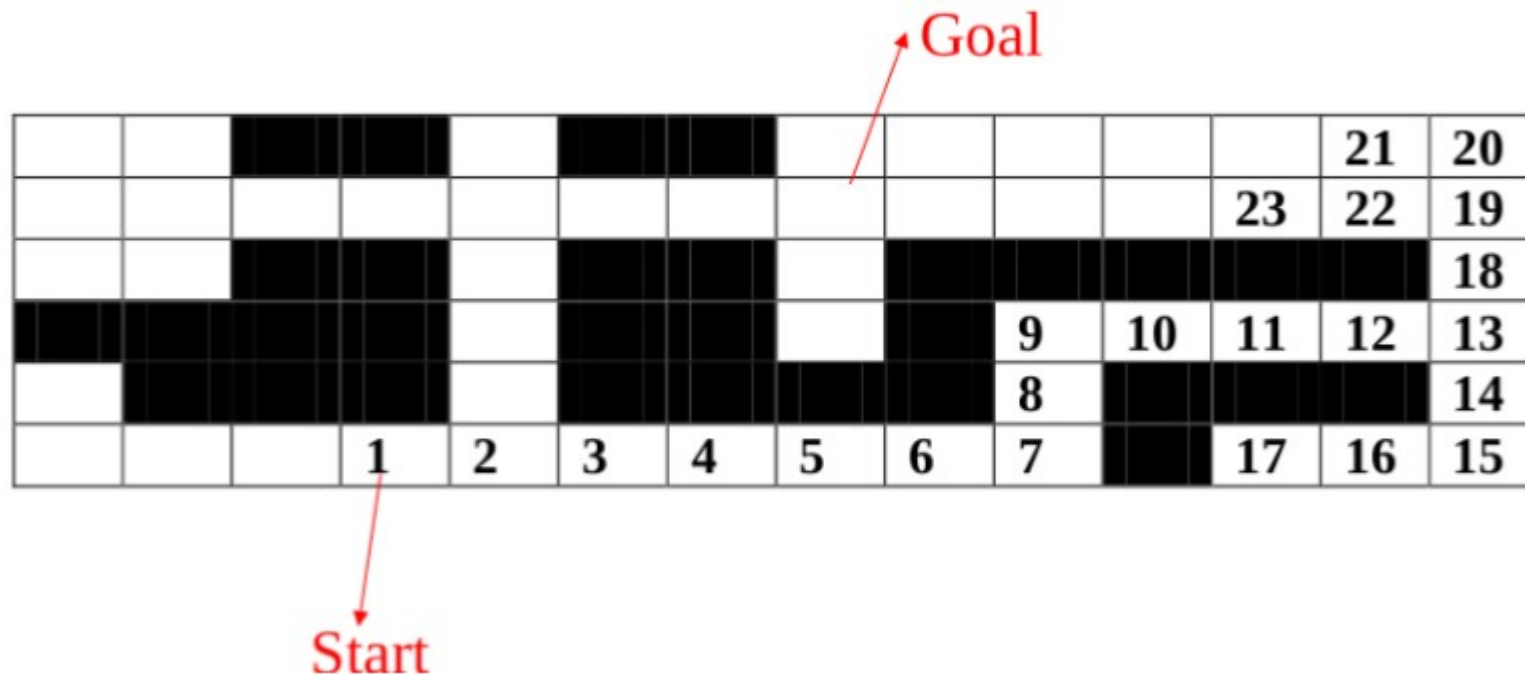
# Depth First Search ( DFS )

- ¿Cómo se desarrolla DFS sobre el mapa?



# Depth First Search ( DFS )

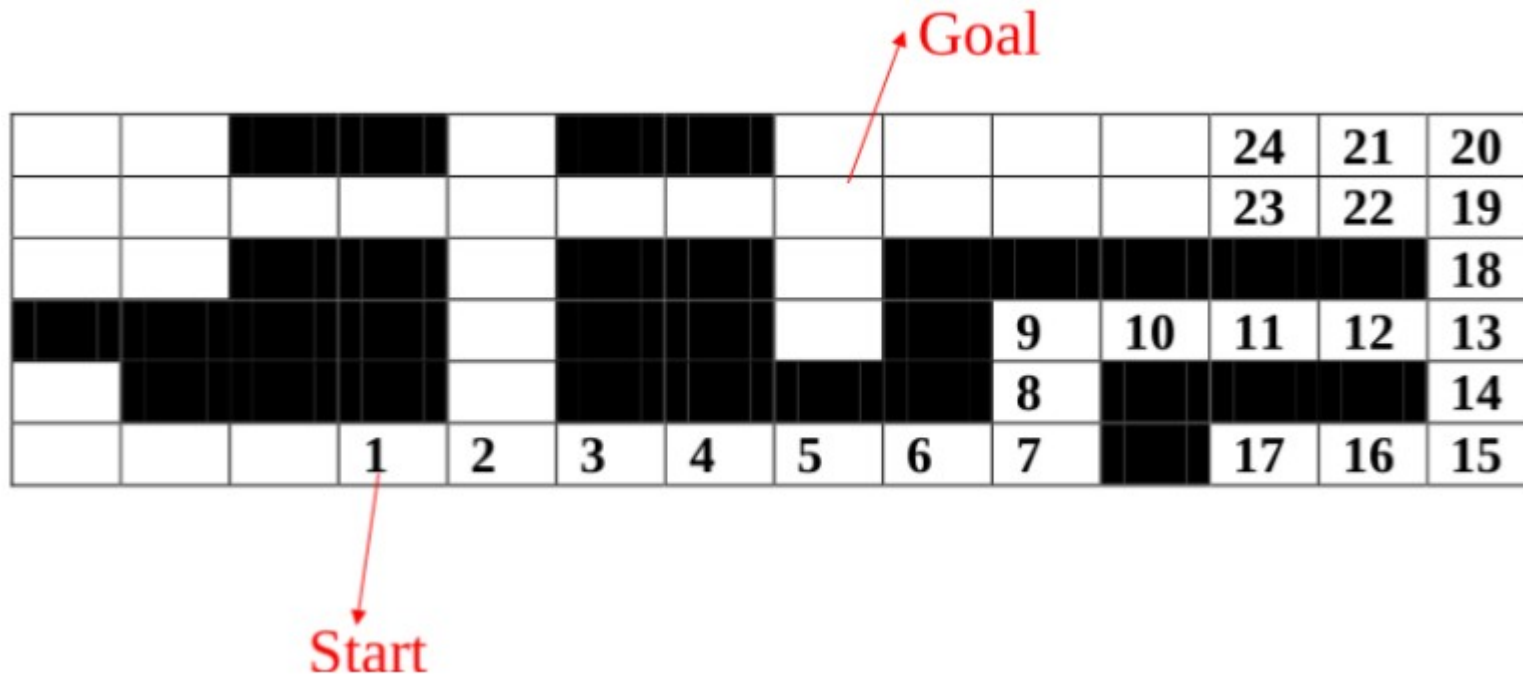
- ¿Cómo se desarrolla DFS sobre el mapa?





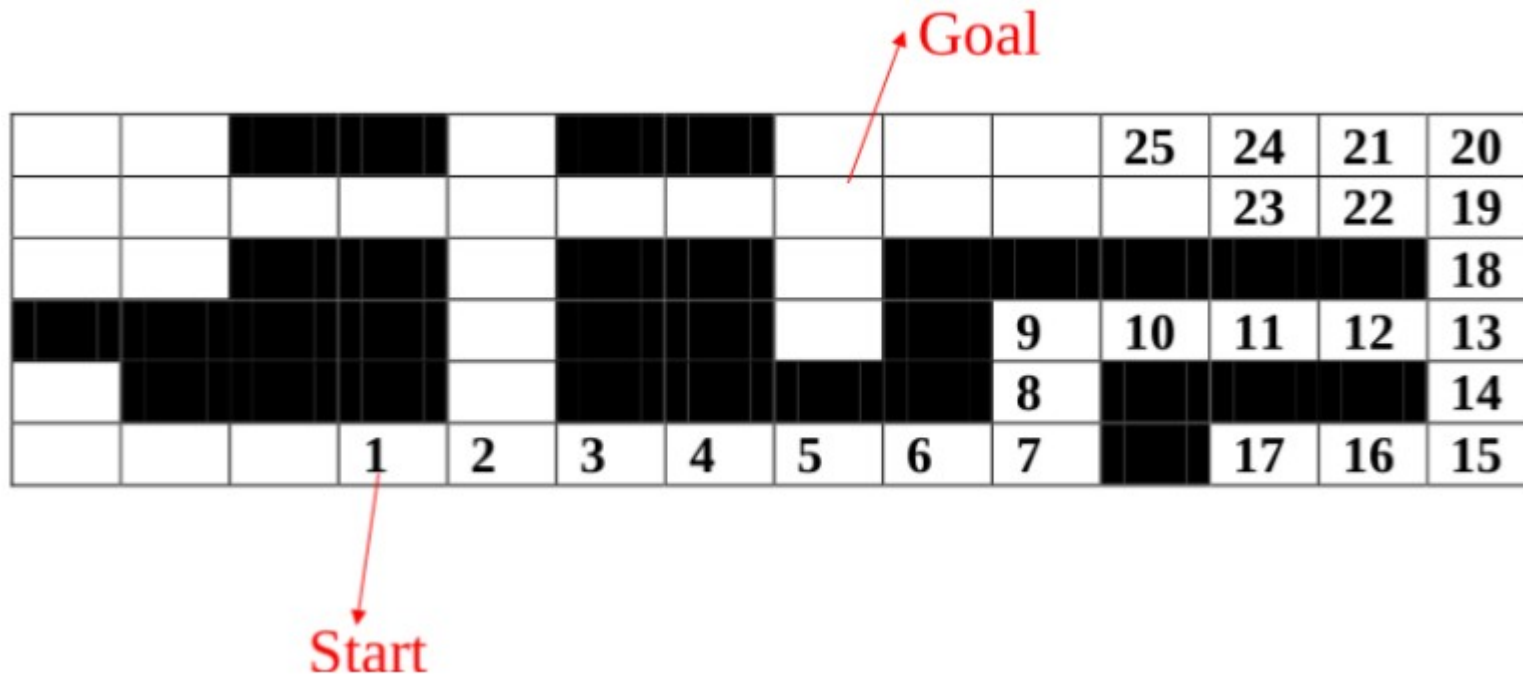
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



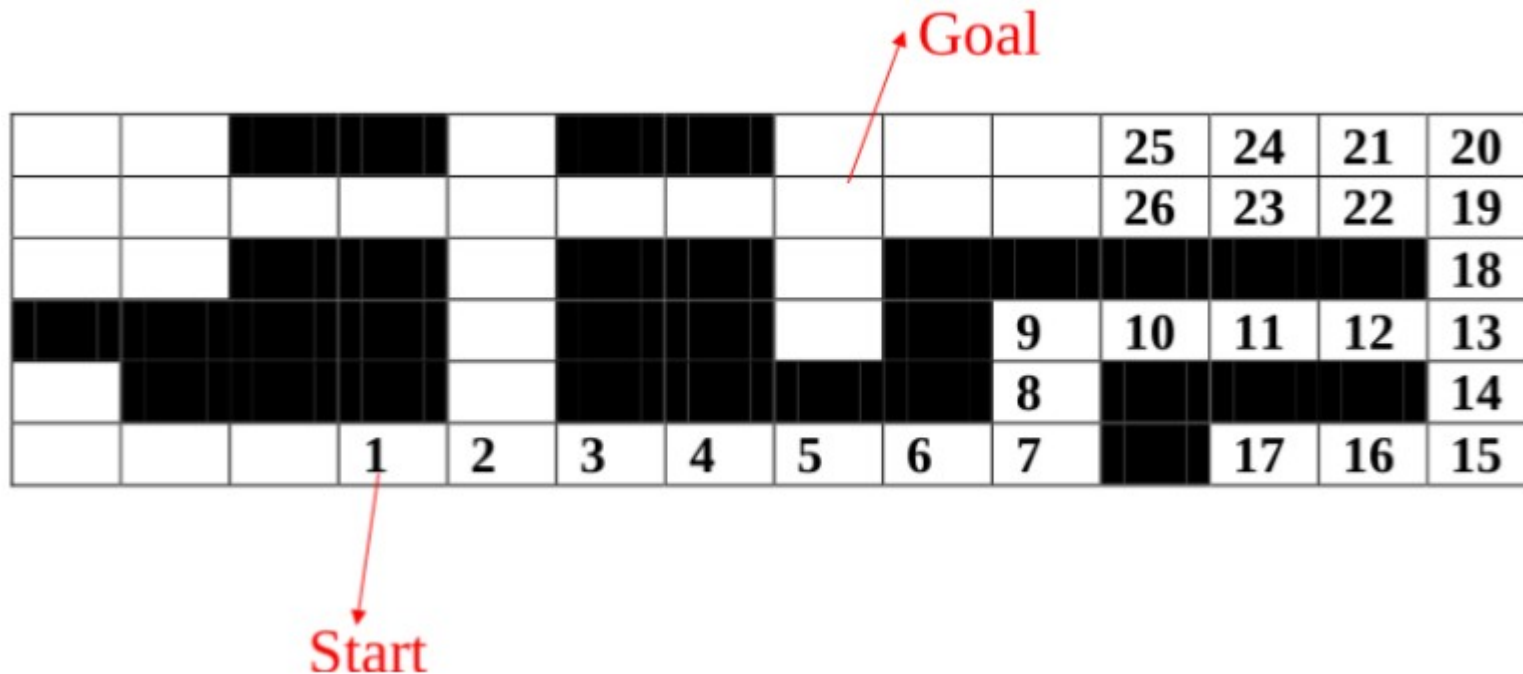
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



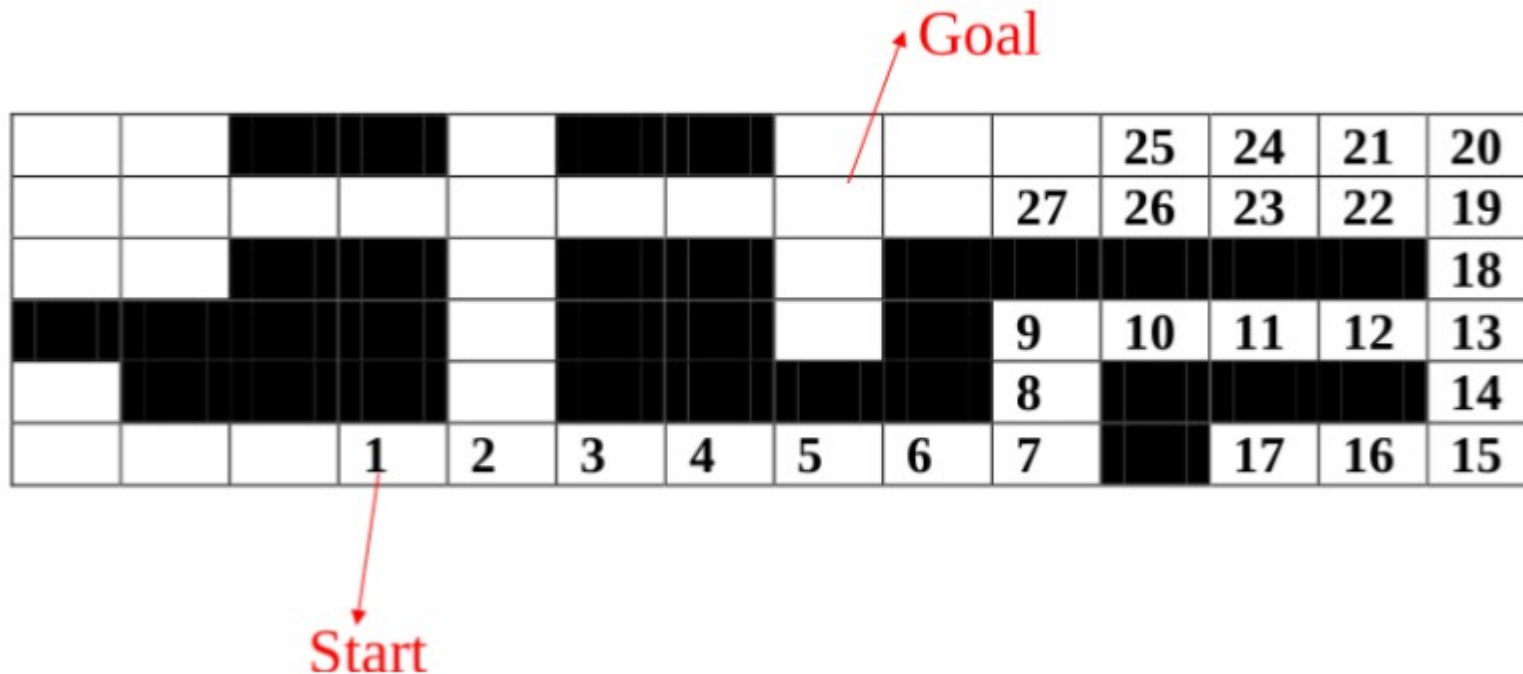
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



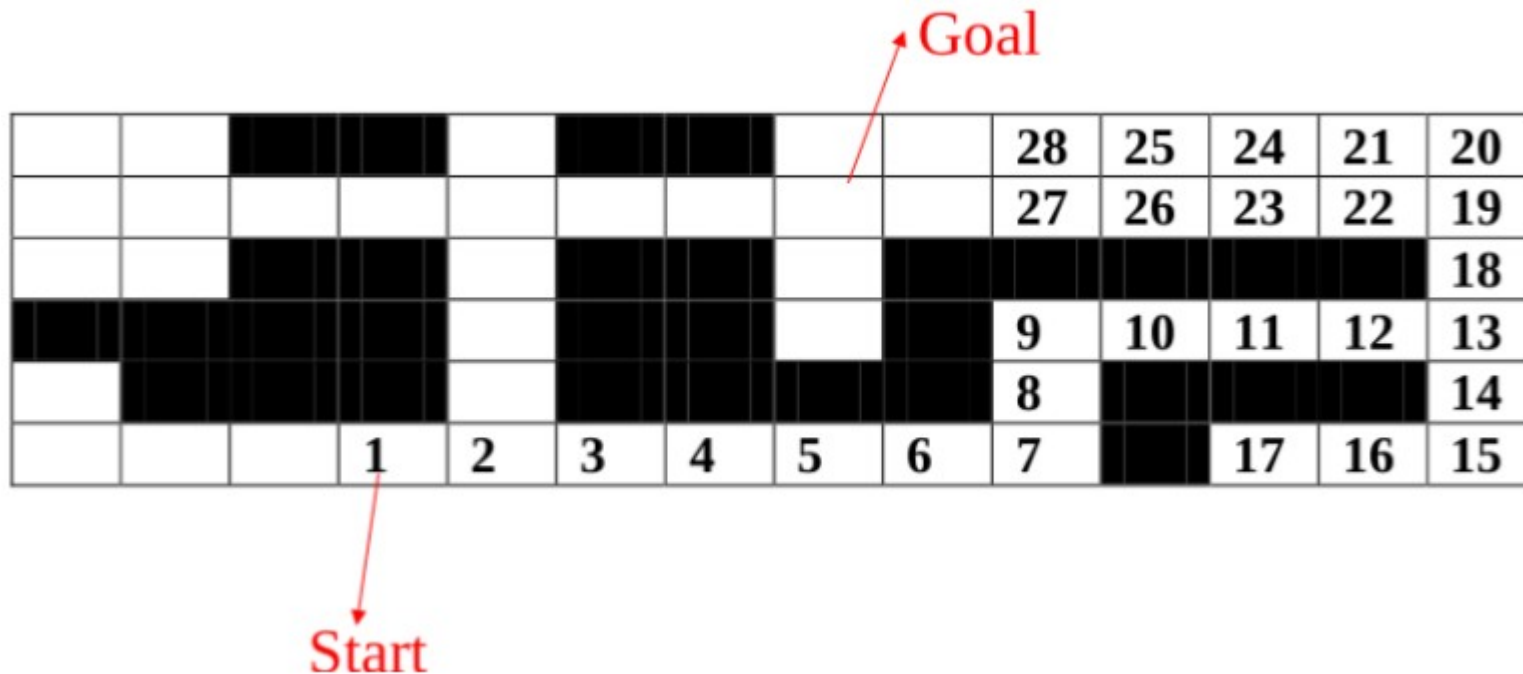
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



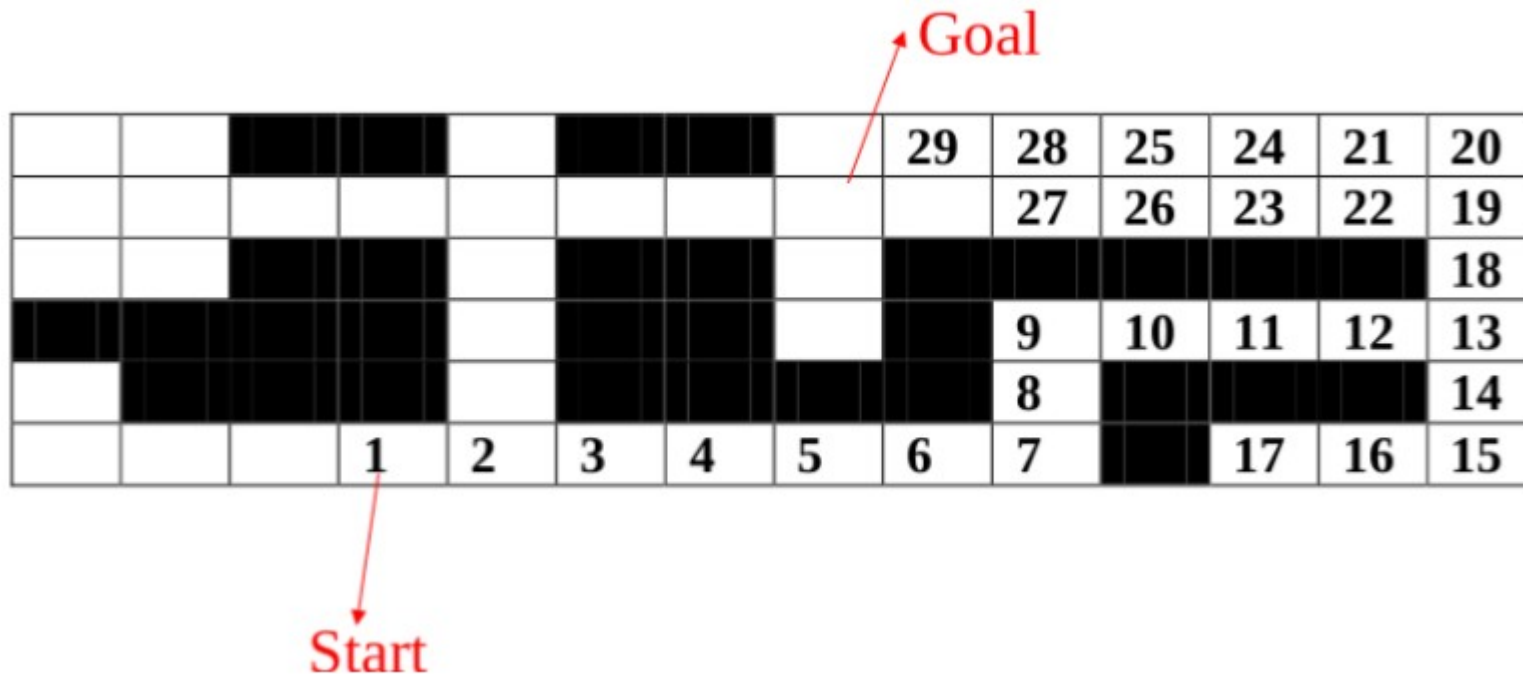
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



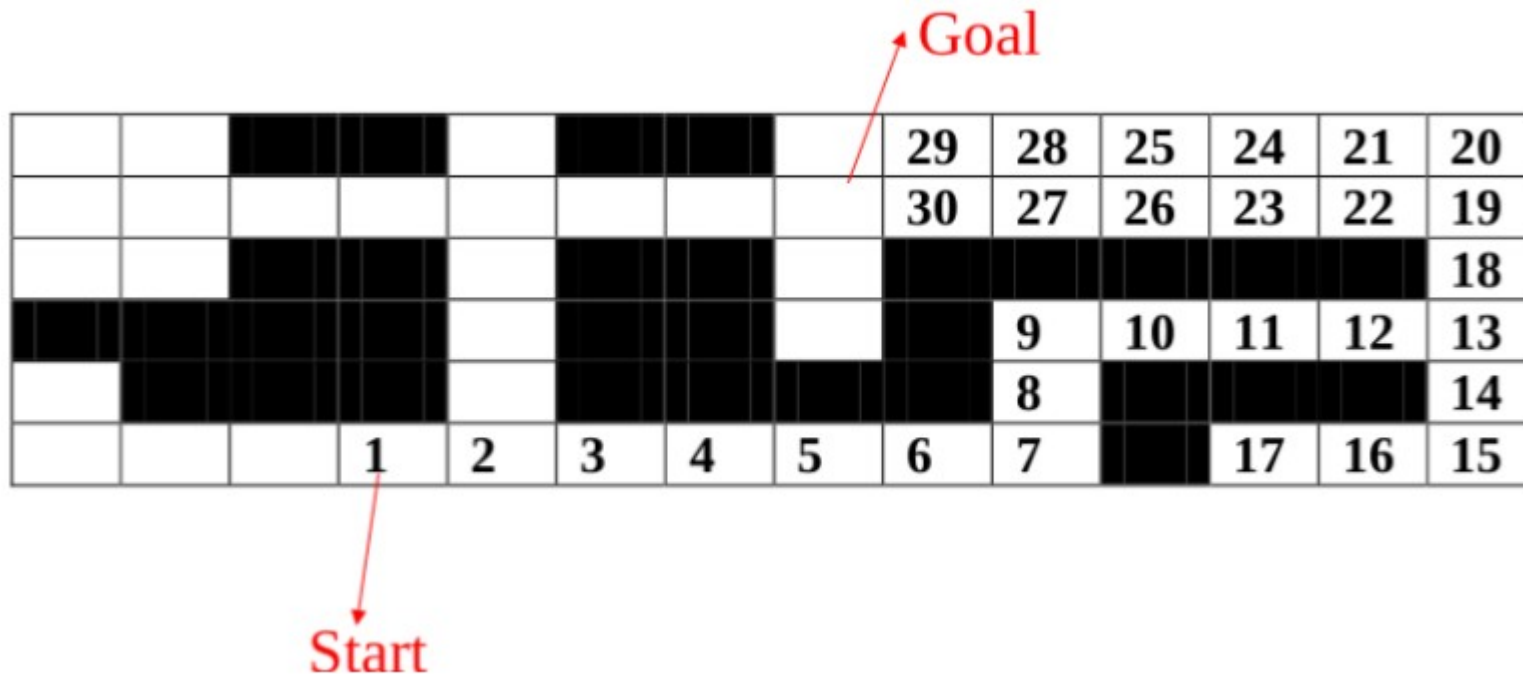
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



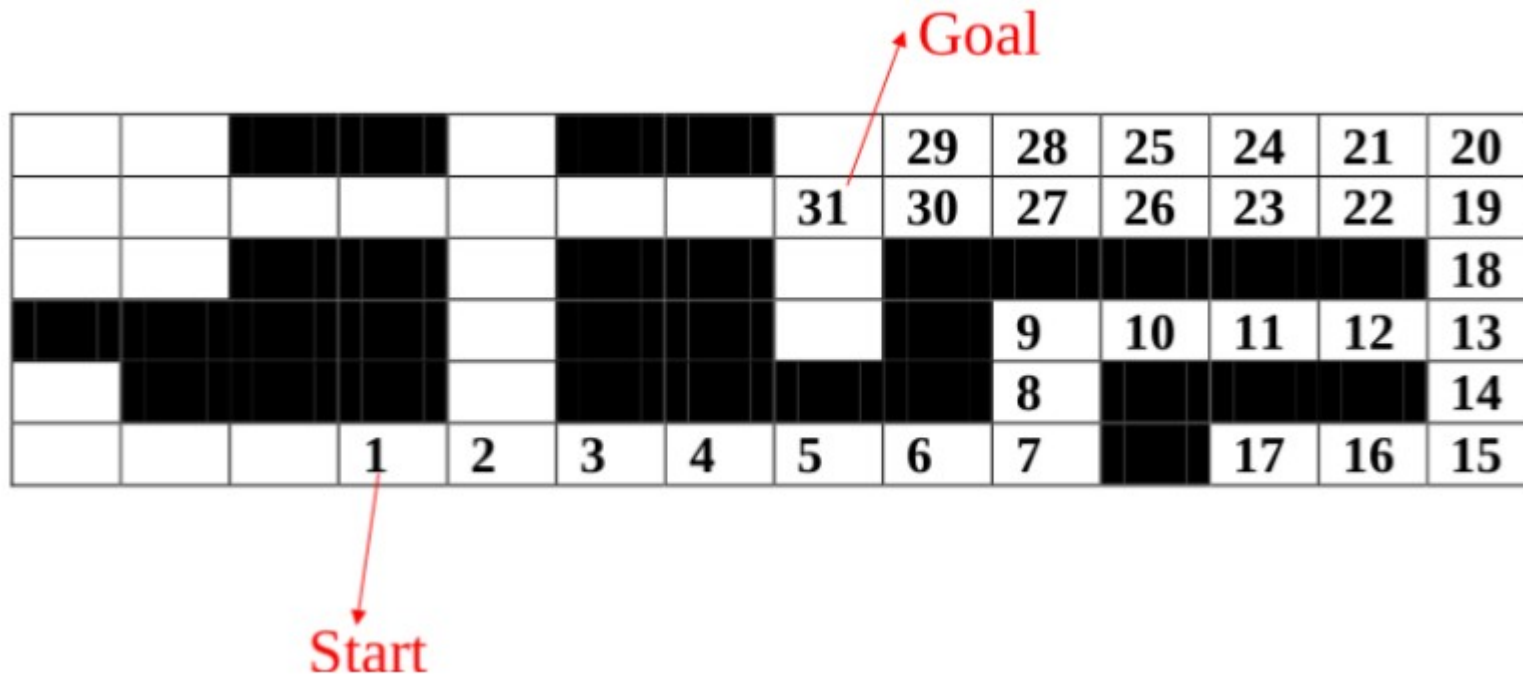
# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?



# Depth First Search ( DFS )

- ¿ Cómo se desarrolla DFS sobre el mapa ?





# Depth First Search ( DFS )

## Algoritmo

```
1: procedure DFS(start_state, goal_state)
2:    $Q \leftarrow \text{Queue}()$ 
3:    $Q.\text{push\_back}(\text{start\_state})$ 
4:   while  $Q$  not empty do
5:      $s = Q.\text{pop\_back}()$                                 ▷ get newer element
6:     if  $s$  is equal  $\text{goal\_state}$  then
7:       return  $s$ 
8:     for all successors in  $s.\text{expand}()$  do                ▷ successors linked to its parent
9:       if successor not discovered then
10:         $\text{successor.set\_discovered}()$ 
11:         $Q.\text{push\_back}(\text{successor})$ 
```

# Bibliografía

- ***Robot Motion Planning***, Latombe, J-C.

Haz un  
paréntesis



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE

RESPONDE  
LA EN(VESTA  
DO(ENTE

Este semestre más que  
nunca tus docentes  
necesitan de tu compromiso