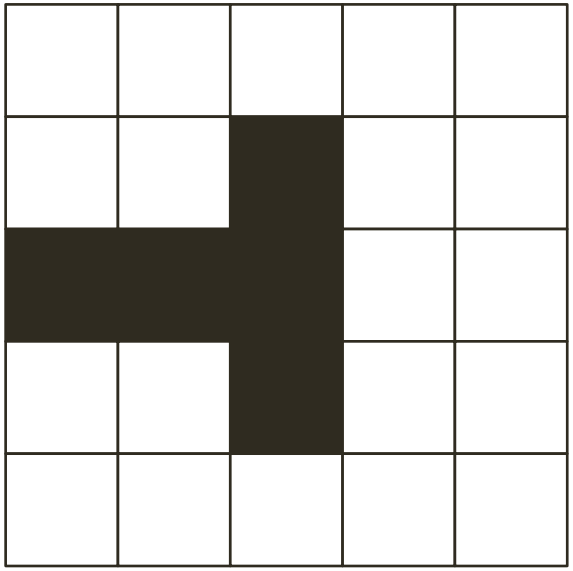
Búsqueda en Anchura y Algoritmo Wavefront

Dr. José de Jesús Hernández Barragán
josed.hernandezb@academicos.udg.mx

Introducción

El algoritmo wavefront encuentra un camino desde un nodo de inicio s a un nodo final g en un mapa con rejillas de ocupación.

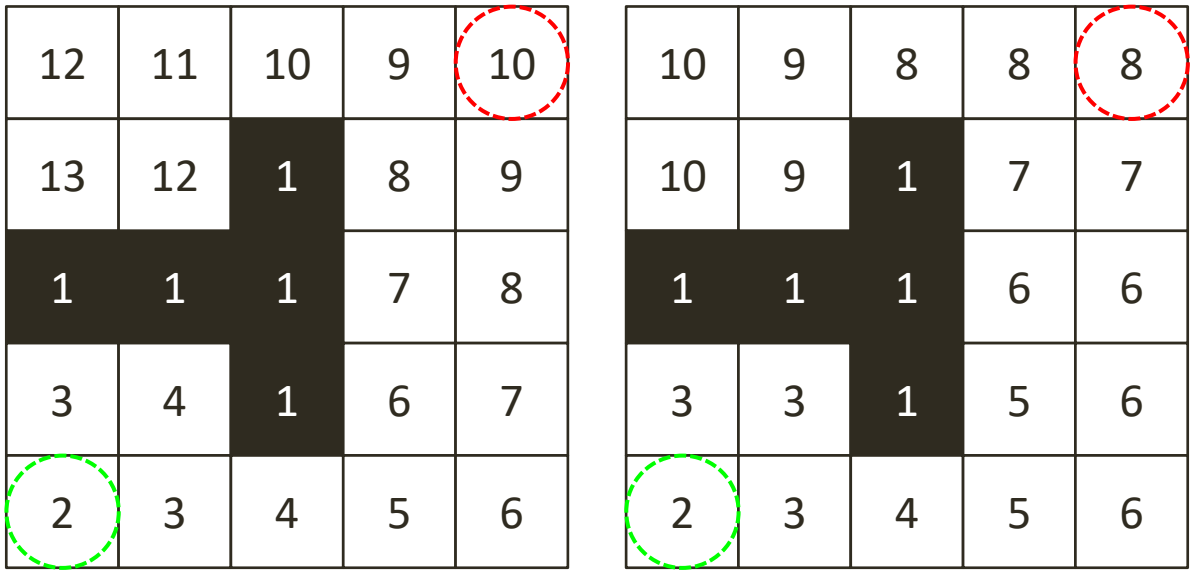


0	0	0	0	s
0	0	1	0	0
1	1	1	0	0
0	0	1	0	0
g	0	0	0	0

Los obstáculos están representados con un valor de 1 y el camino transitable con un 0.

Introducción (continuación)

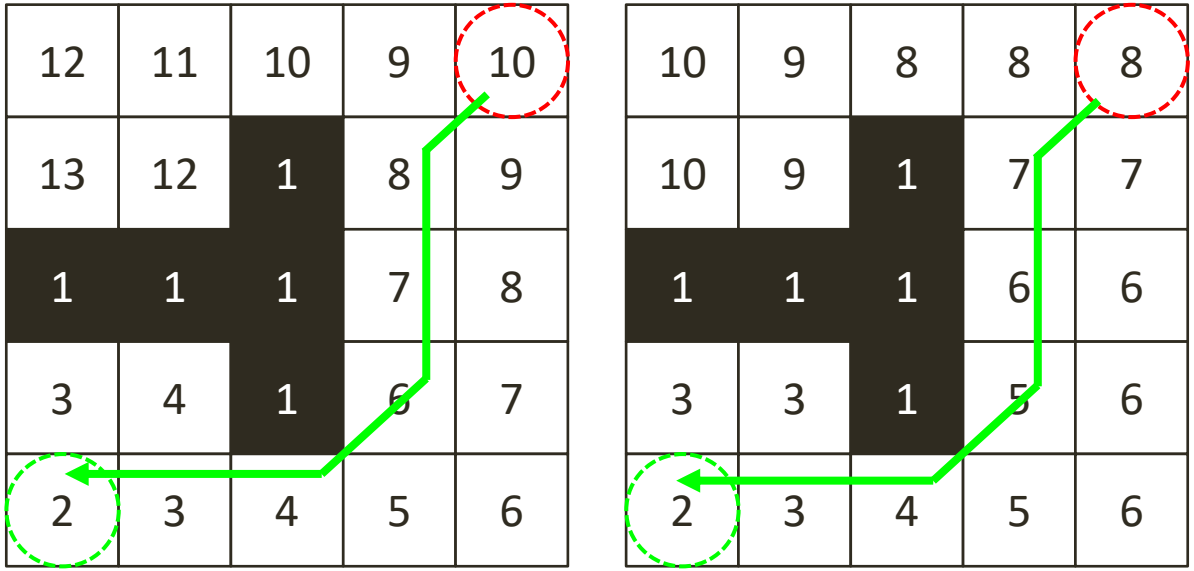
El algoritmo utiliza una búsqueda en anchura (BFS) empezando en el nodo final g con un valor inicial de 2.



Mientras que la BFS recorre el mapa, a cada nodo se agina un valor que corresponde a la suma acumulativa del coste del camino más corto desde el nodo actual al nodo g.

Introducción (continuación)

Para encontrar el camino más corto, se empieza en el nodo s y se recorre cada nodo con menor coste hasta llegar al nodo g.



Conexión entre rejillas

En un mapa representado en rejillas de ocupación, existen dos maneras de conectar nodos: la vecindad de Neumann y la vecindad de Moore.

	1	
4	C	2
	3	

En la vecindad de Neumann, el nodo en la celda C se conecta con los 4 nodos adyacentes a las caras de C.

8	1	2
7	C	3
6	5	4

En la vecindad de Moore, el nodo en la celda C se conecta con los 4 nodos adyacentes a las caras de C y a los 4 nodos de cada esquina.

Búsqueda en anchura

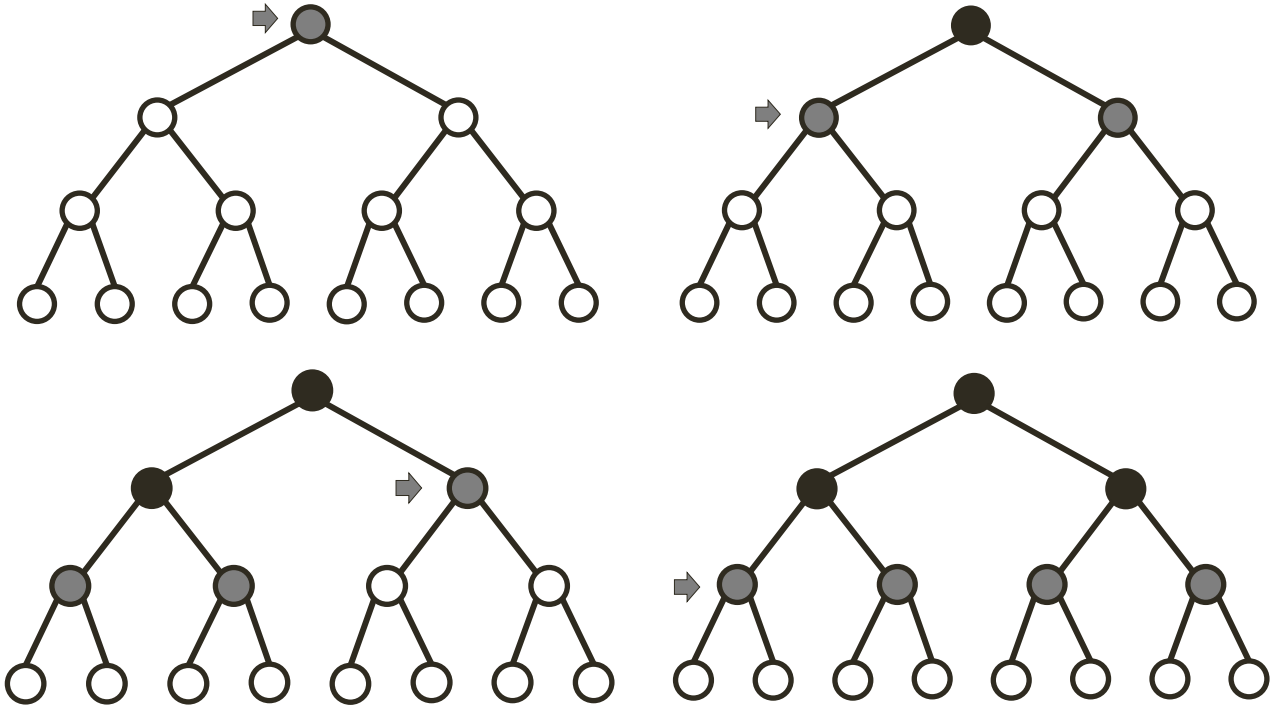
En la búsqueda en anchura se utiliza para recorrer o buscar elementos en un grafo.

El algoritmo primero explora los nodos superficiales, es decir, los nodos mas cercanos al nodo inicial.

- En una lista cerrada c se guardan los nodos visitados.
- En una lista abierta a se agregar nuevos nodos al final, y los nodos al inicio se seleccionan para continuar con la exploración.

Búsqueda en anchura (continuación)

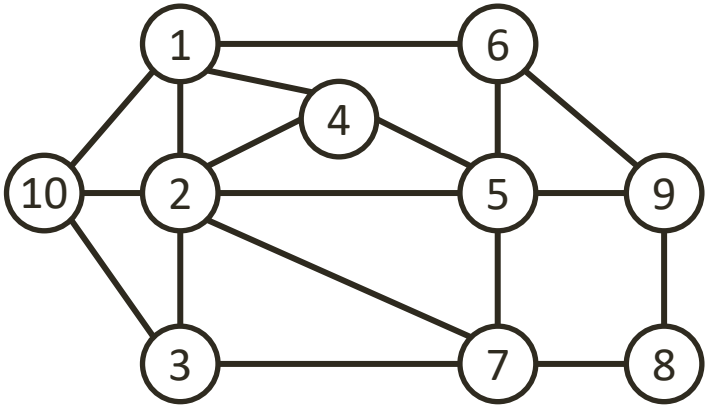
Ejemplo el proceso de búsqueda:



- El nodo actual se representa con una flecha.
- Los nodos en la lista abierta están en color gris.
- Los nodos en la lista cerrada están en negro.
- Los nodos sin visitar están en blanco.

Búsqueda en anchura (continuación)

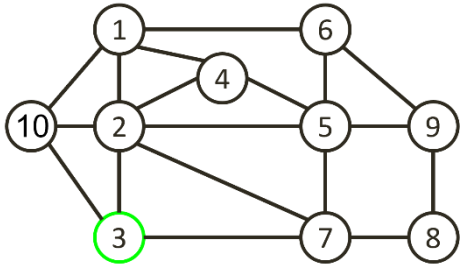
Ejemplo en grafo: Encontrar el recorrido de un nodo inicial a todos los nodos del siguiente grafo.



El nodo inicial se selección con el número 3. Se recomienda llevar el recorrido con una lista para nodos padres s y una lista para nodos hijos t.

Búsqueda en anchura (continuación)

Ejemplo en grafo:



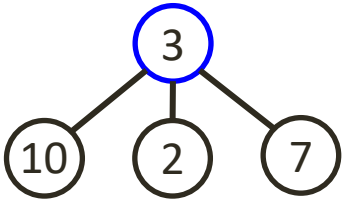
Paso 1:

Al inicio la lista de cerrados inicia vacía

$a = [3]$
$c = [\ \ ]$

$s = [\ \ ]$
$t = [\ \ ]$

Paso 2:

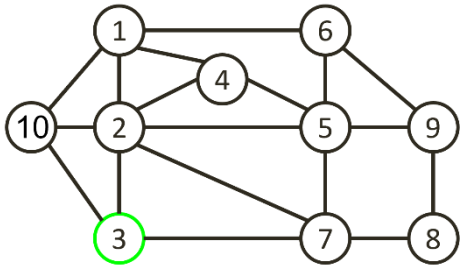


$a = [10 \quad 2 \quad 7]$
$c = [3]$

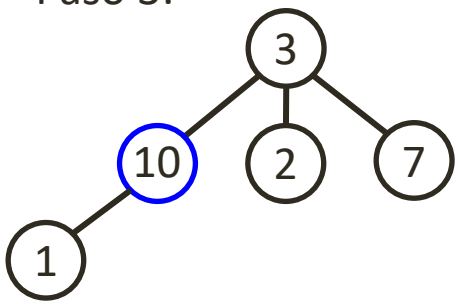
$s = [3 \quad 3 \quad 3]$
$t = [10 \quad 2 \quad 7]$

Búsqueda en anchura (continuación)

Ejemplo en grafo:



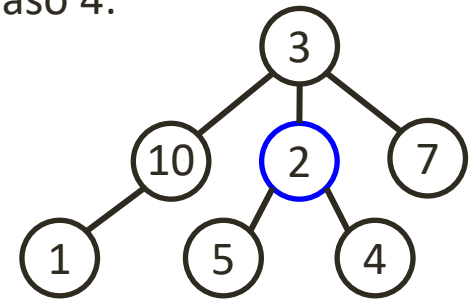
Paso 3:



$$a = [2 \quad 7 \quad 1]$$
$$c = [3 \quad 10]$$

$$s = [3 \quad 3 \quad 3 \quad 10]$$
$$t = [10 \quad 2 \quad 7 \quad 1]$$

Paso 4:

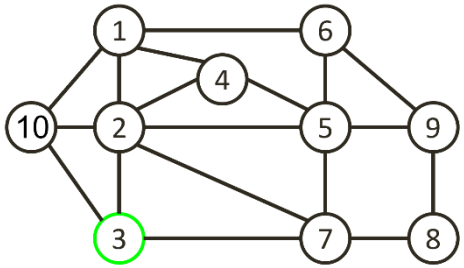


$$a = [7 \quad 1 \quad 5 \quad 4]$$
$$c = [3 \quad 10 \quad 2]$$

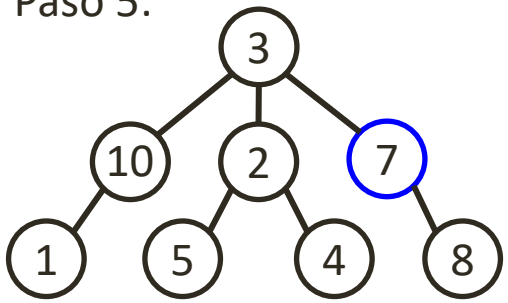
$$s = [3 \quad 3 \quad 3 \quad 10 \quad 2 \quad 2]$$
$$t = [10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4]$$

Búsqueda en anchura (continuación)

Ejemplo en grafo:



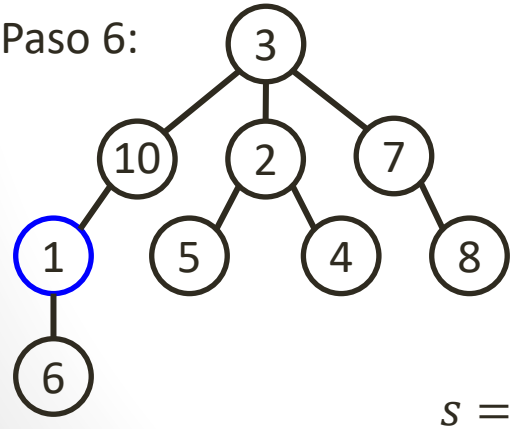
Paso 5:



$$a = [1 \quad 5 \quad 4 \quad 8]$$
$$c = [3 \quad 10 \quad 2 \quad 7]$$

$$s = [3 \quad 3 \quad 3 \quad 10 \quad 2 \quad 2 \quad 7]$$
$$t = [10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 8]$$

Paso 6:

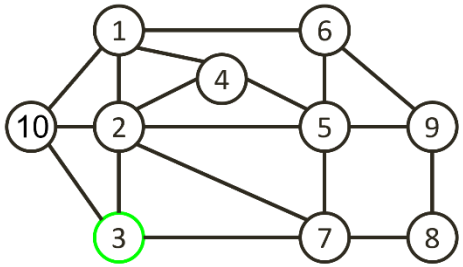


$$a = [5 \quad 4 \quad 8 \quad 6]$$
$$c = [3 \quad 10 \quad 2 \quad 7 \quad 1]$$

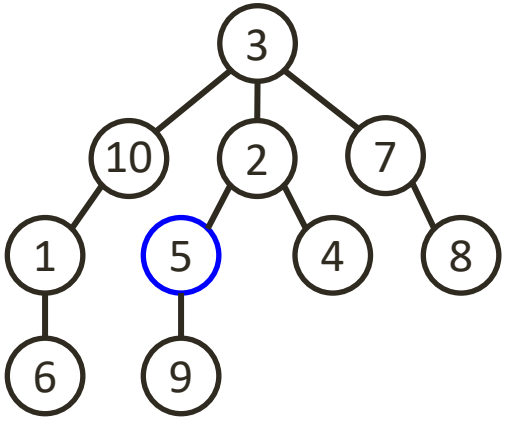
$$s = [3 \quad 3 \quad 3 \quad 10 \quad 2 \quad 2 \quad 7 \quad 1]$$
$$t = [10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 8 \quad 6]$$

Búsqueda en anchura (continuación)

Ejemplo en grafo:



Paso 7:

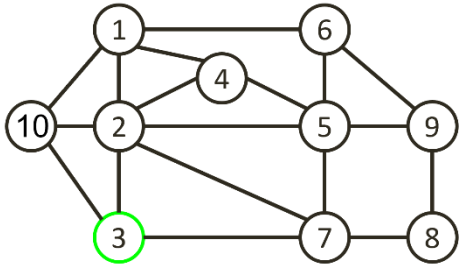


$$a = [4 \quad 8 \quad 6 \quad 9]$$
$$c = [3 \quad 10 \quad 2 \quad 7 \quad 1 \quad 5]$$

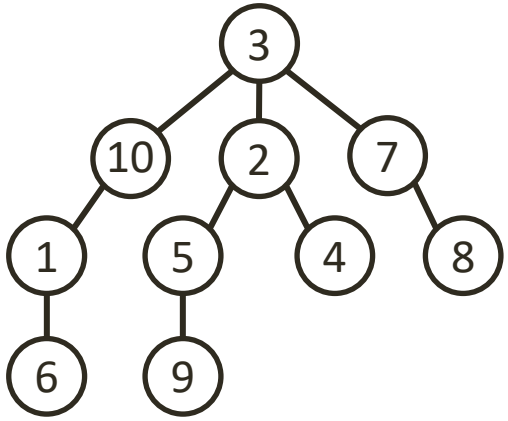
$$s = [3 \quad 3 \quad 3 \quad 10 \quad 2 \quad 2 \quad 7 \quad 1 \quad 5]$$
$$t = [10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 8 \quad 6 \quad 9]$$

Búsqueda en anchura (continuación)

Ejemplo en grafo:



Pasos finales:



$$s = [3 \quad 3 \quad 3 \quad 10 \quad 2 \quad 2 \quad 7 \quad 1 \quad 5]$$
$$t = [10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 8 \quad 6 \quad 9]$$

$$a = [8 \quad 6 \quad 9]$$
$$c = [3 \quad 10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4]$$

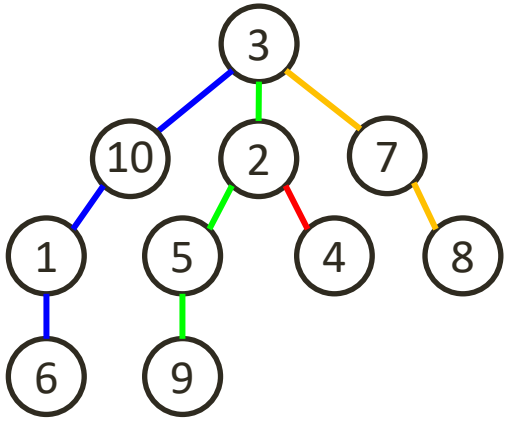
$$a = [6 \quad 9]$$
$$c = [3 \quad 10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 8]$$

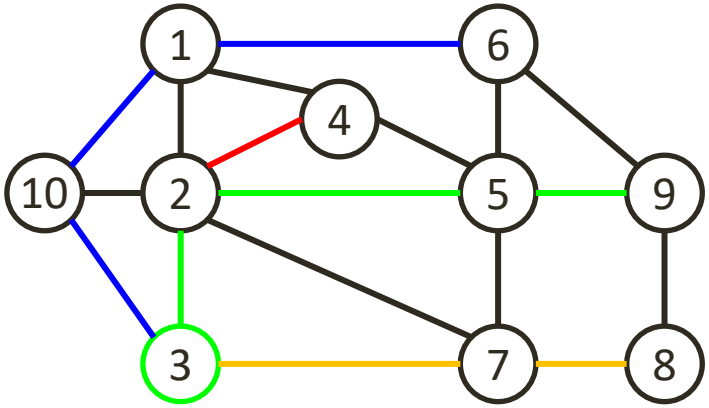
$$a = [9]$$
$$c = [3 \quad 10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 6]$$

$$a = [\ ]$$
$$c = [3 \quad 10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 9]$$

Búsqueda en anchura (continuación)

Ejemplo en grafo: finalmente se muestran los recorridos en el grafo.

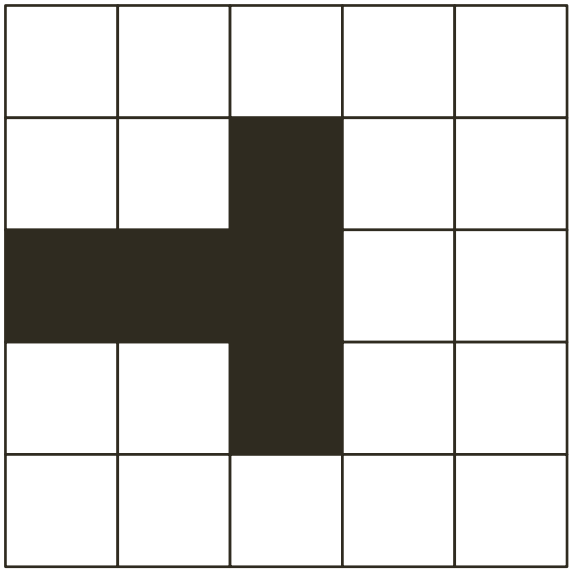




$s = [3 \quad 3 \quad 3 \quad 10 \quad 2 \quad 2 \quad 7 \quad 1 \quad 5]$
$t = [10 \quad 2 \quad 7 \quad 1 \quad 5 \quad 4 \quad 8 \quad 6 \quad 9]$

Búsqueda en anchura (continuación)

La implementación es similar en una mapa representado con rejillas de ocupación.

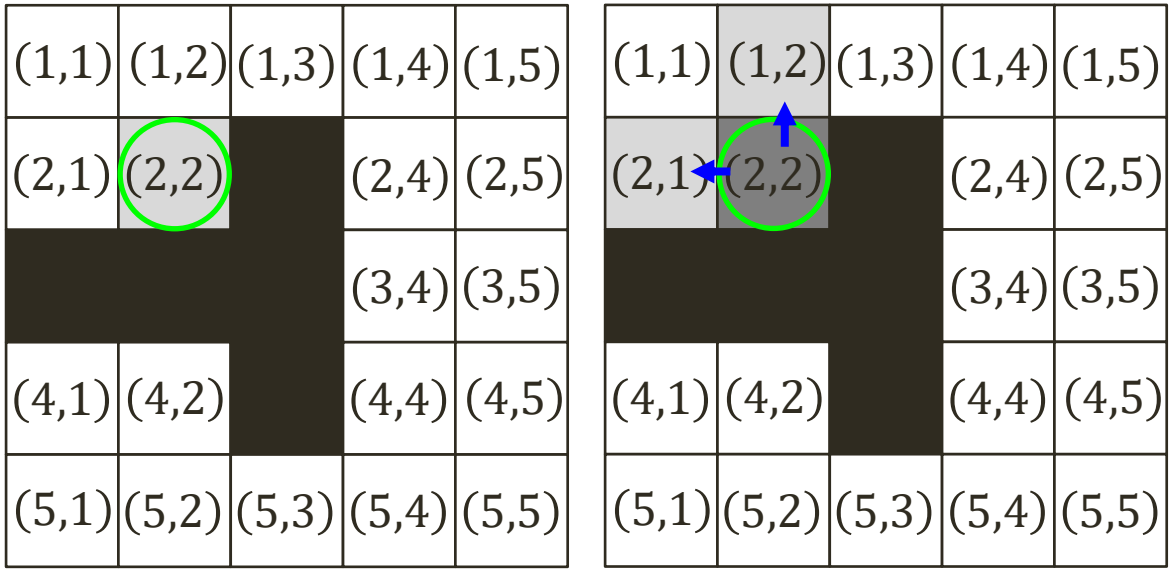


(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)		(2,4)	(2,5)
			(3,4)	(3,5)
(4,1)	(4,2)		(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

Se selecciona una rejilla inicial y se realiza el recorrido a todas las rejillas del mapa. También se tiene que definir la conexión entre rejillas.

Búsqueda en anchura (continuación)

Ejemplo en rejillas de ocupación (vecindad de Neumann):



$$s = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

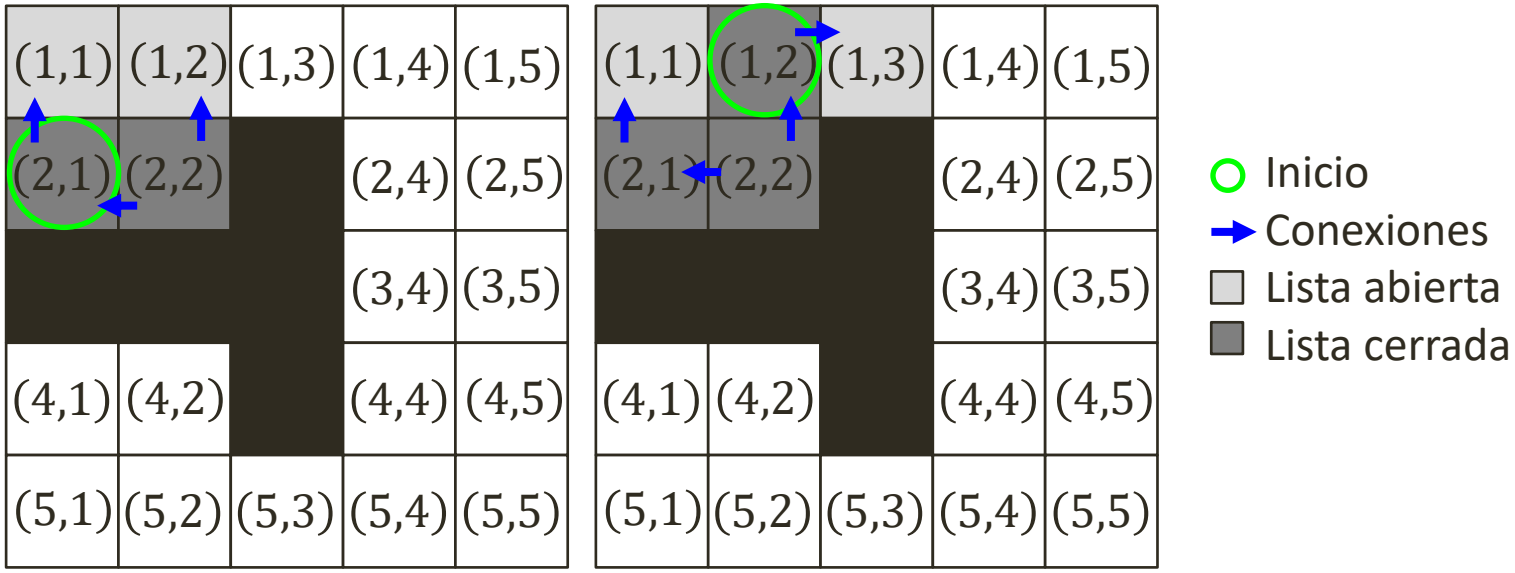
$$t = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

No se realizan conexiones en rejillas ocupadas o fuera de los limites del mapa.

Ahora las listas representan matrices, donde cada columna indica el valor de una rejilla.

Búsqueda en anchura (continuación)

Ejemplo en rejillas de ocupación (vecindad de Neumann):

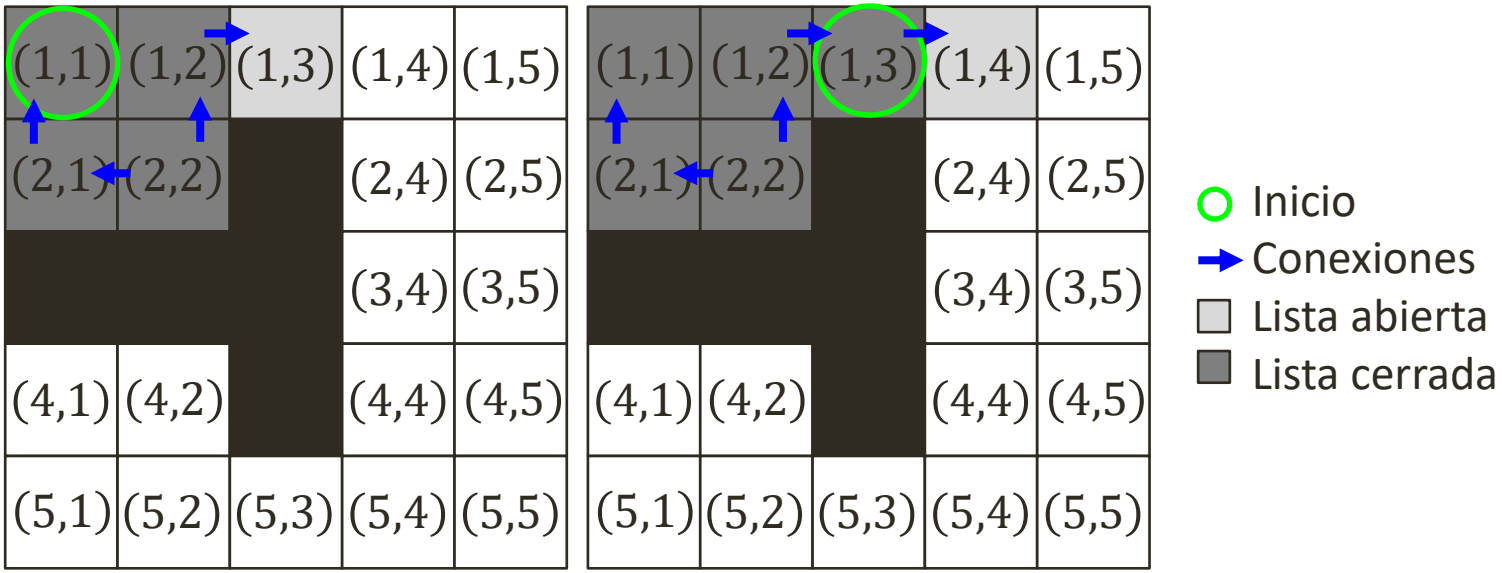


$$s = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 2 \end{bmatrix}$$

$$t = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 3 \end{bmatrix}$$

Búsqueda en anchura (continuación)

Ejemplo en rejillas de ocupación (vecindad de Neumann):

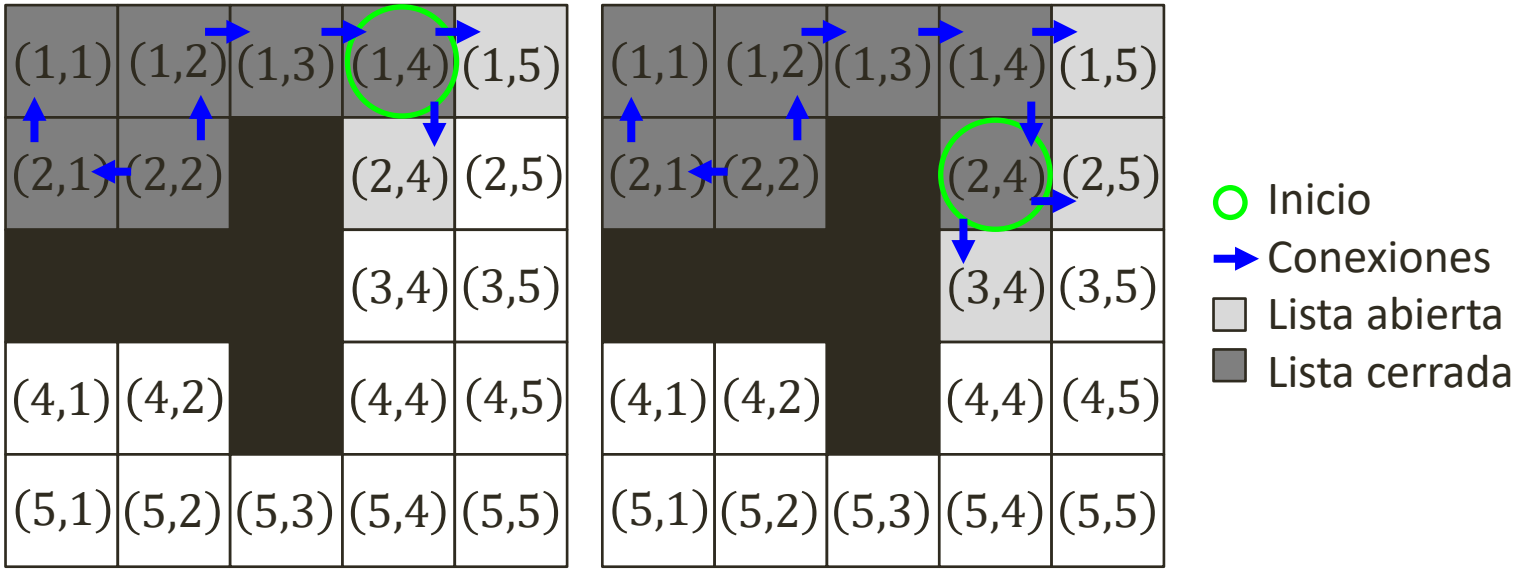


$$s = \begin{bmatrix} 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 2 & 3 \end{bmatrix}$$

$$t = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 3 & 4 \end{bmatrix}$$

Búsqueda en anchura (continuación)

Ejemplo en rejillas de ocupación (vecindad de Neumann):

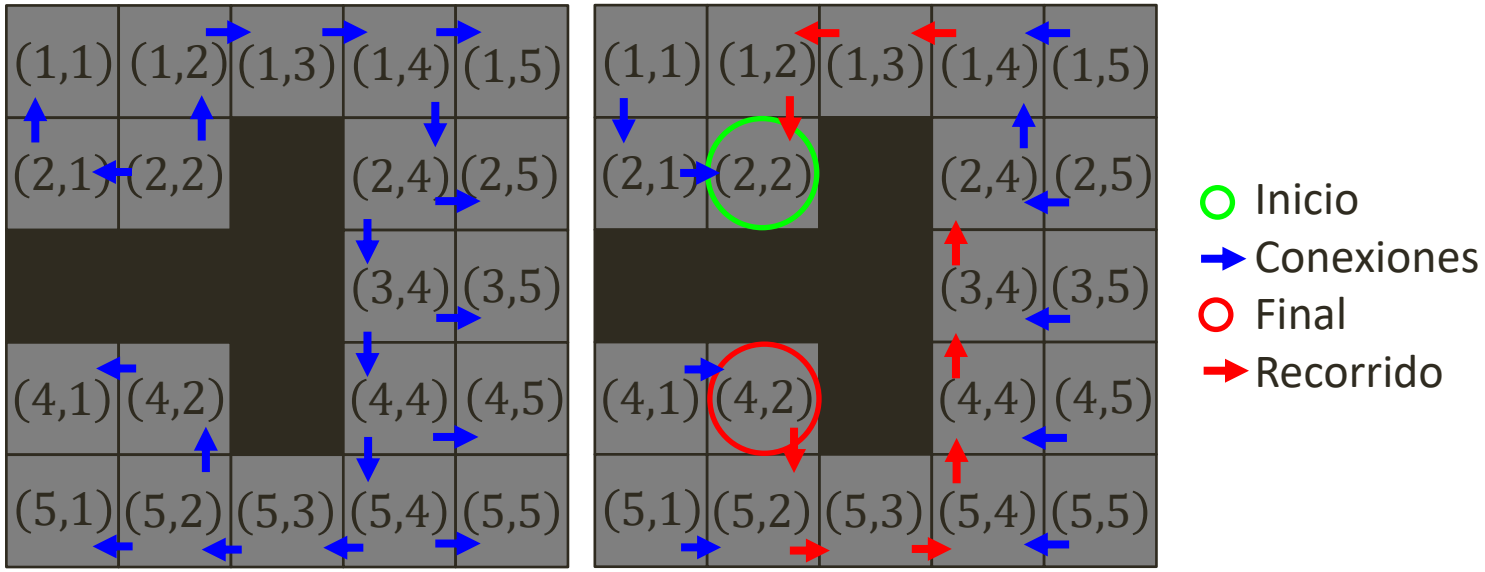


$$s = \begin{bmatrix} 2 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 2 \\ 2 & 2 & 1 & 2 & 3 & 4 & 4 & 4 & 4 \end{bmatrix}$$

$$t = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 3 \\ 1 & 2 & 1 & 3 & 4 & 5 & 4 & 5 & 4 \end{bmatrix}$$

Búsqueda en anchura (continuación)

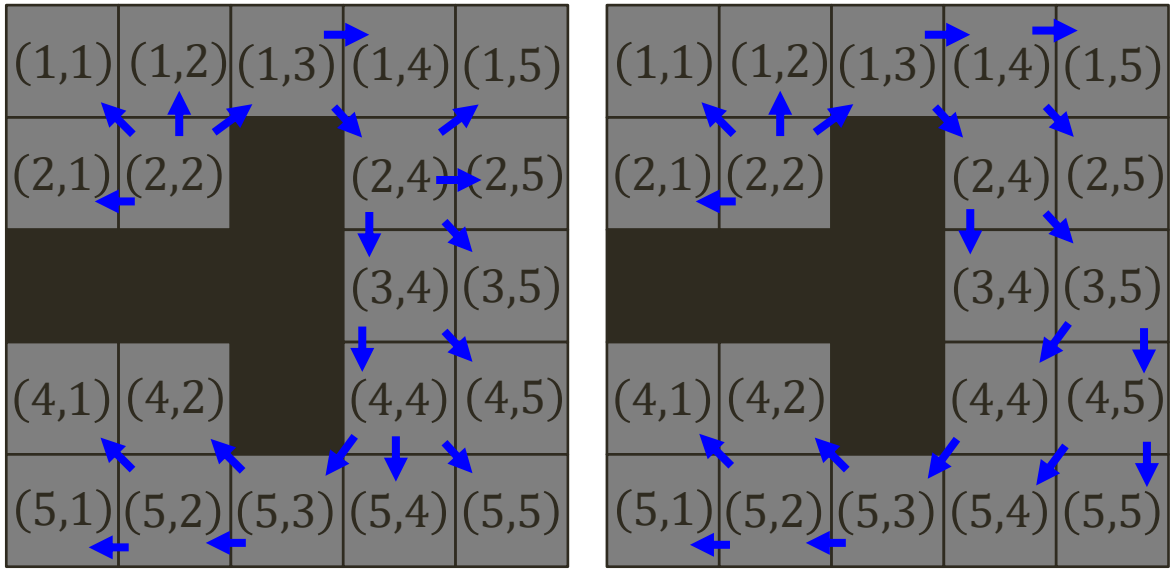
Ejemplo en rejillas de ocupación (vecindad de Neumann): Se realiza el recorrido por todas las rejillas en el mapa.



Después, se selecciona una rejilla final y se crea una trayectoria con el recorrido de la BFS.

Búsqueda en anchura (continuación)

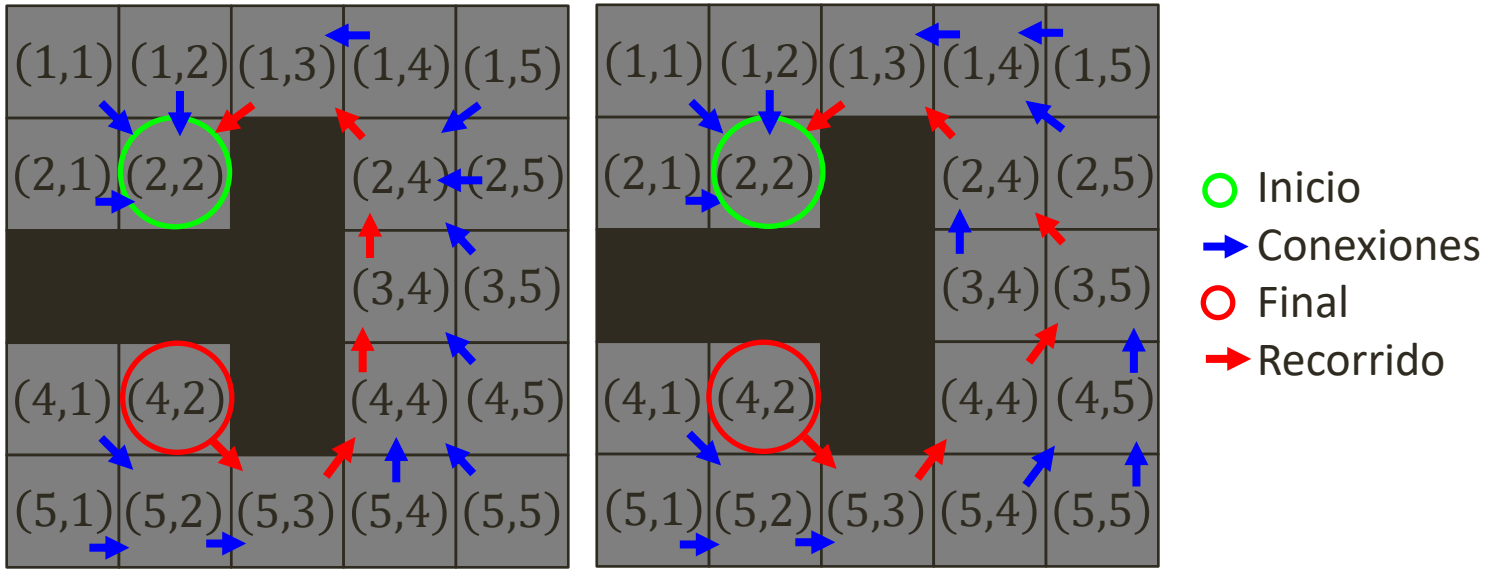
Ejemplo en rejillas de ocupación (vecindad de Moore): Se realiza el recorrido por todas las rejillas en el mapa.



Las conexiones entre rejillas cambias dependiendo de orden de los elementos en la lista abierta.

Búsqueda en anchura (continuación)

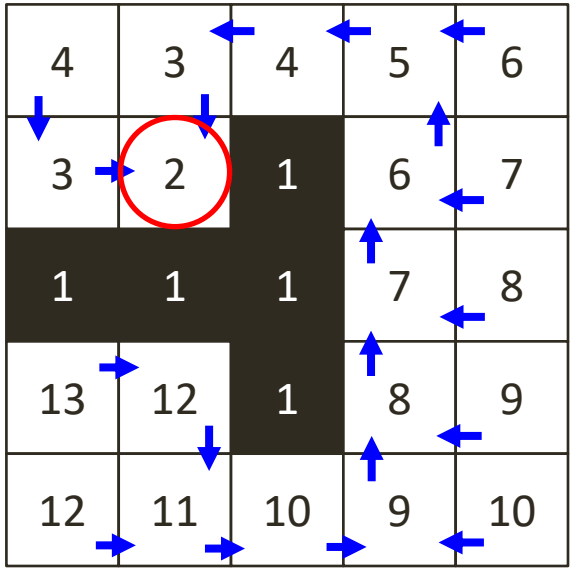
Después, se selecciona una rejilla final y se crea una trayectoria con el recorrido de la BFS.



La trayectoria depende de las conexiones entre rejillas y no siempre representa la trayectoria más corta.

Algoritmo Wavefront

El algoritmo de wavefront toma como base el recorrido de la BFS para asignar un peso a cada rejilla.



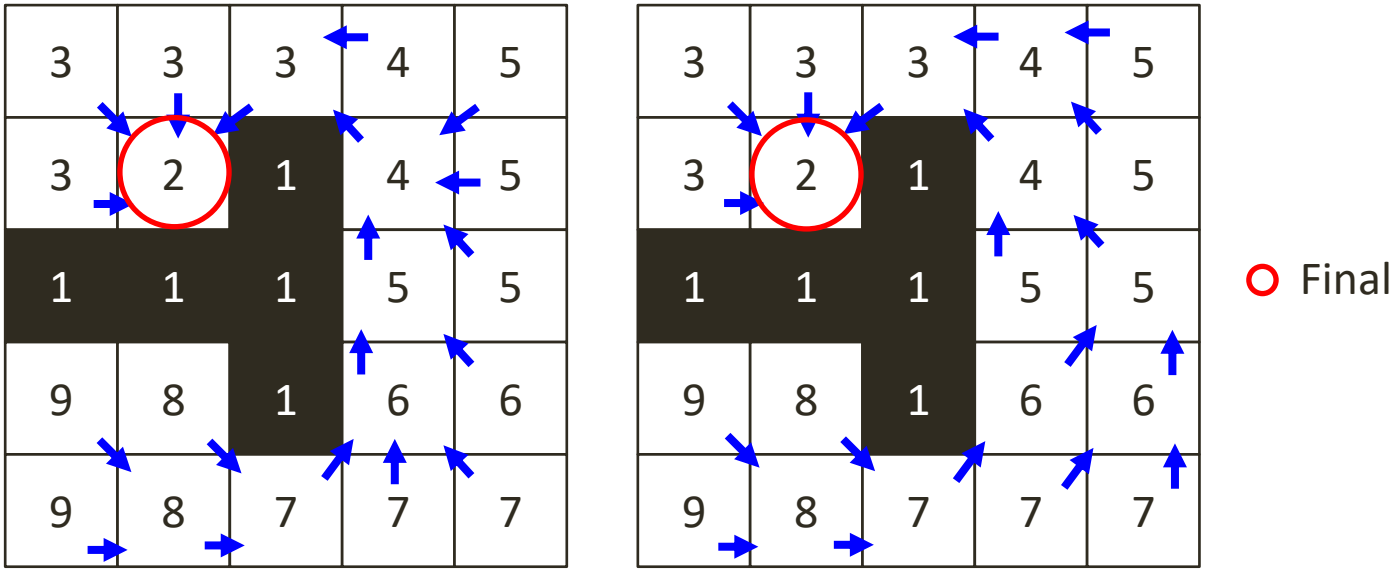
Ejemplo para un recorrido con la vecindad de Neumann.

Final

El peso de cada rejilla corresponde al costo del recorrido de la rejilla actual a la rejilla inicial.

Algoritmo Wavefront (continuación)

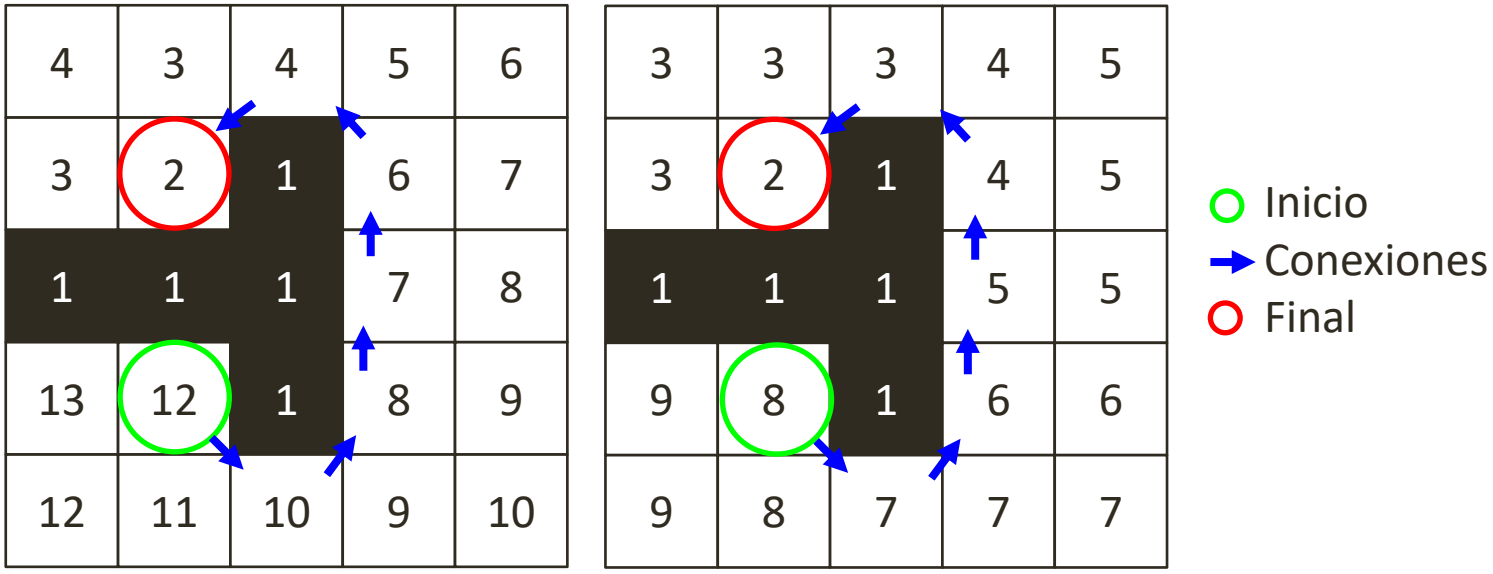
Ejemplo para un recorrido con la vecindad de Moore.



Se muestran los recorridos de dos soluciones de la BFS. Sin embargo, los pesos del algoritmo de wavefront son los mismos.

Algoritmo Wavefront (continuación)

Para encontrar el camino más corto, se define la rejilla de inicio y se recorre cada rejilla con menor coste hasta llegar a la rejilla final.

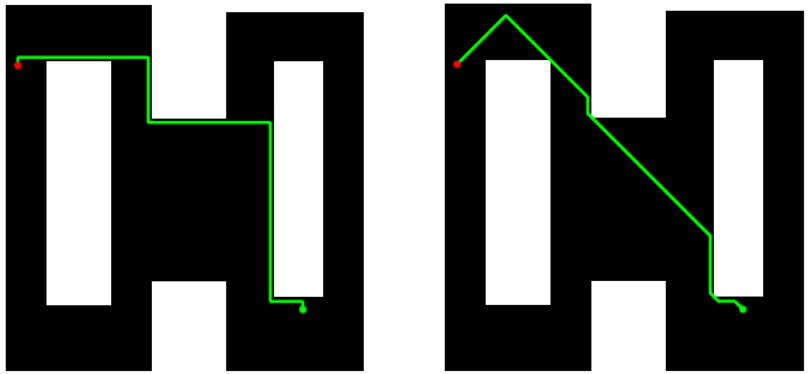


Con la vecindad de Neumann

Con la vecindad de Moore

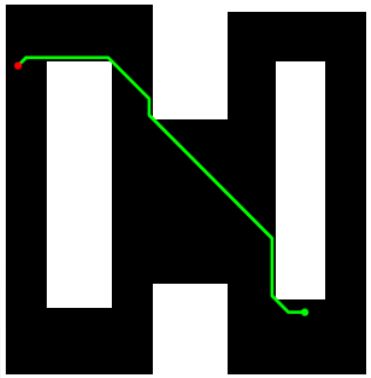
Algoritmo Wavefront (continuación)

Implementaciones del algoritmo BFS y el algoritmo wavefront.

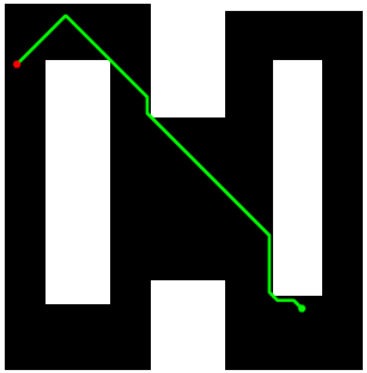


BFS - Neumann

BFS - Moore



WF - Neumann



WF - Moore

- Rejilla libre
- Obstáculos
- Final
- Trayectoria
- Inicio

BFS – Búsqueda en anchura
WF – Algoritmo wavefront