

# EVIDENCIAS DE ALGORITMO WAVEFRONT

Como objetivo principal, se inicializa el programa con un menú que permite al usuario decidir cual es la vecindad que desea usar. En este caso, el programa posee dos vecindades (**Neuman y Moore**).

A continuación, se hace presente la evidencia sobre el funcionamiento del programa.

```
1. John von Vecindades
2. Edward Forrest Moore
3. Salir
Elige una opcion
Introduce un numero entero: 1
```

```
Fila posición inicial: 3
Columna posición inicial: 2
Fila posición final: 3
Columna posición final: 4
```

Matriz Original

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  0]
 [ 0  0  0 -1  0]
 [ 0  0  0 -1  0]]
```

Matriz iniciada

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  0]
 [ 0  0  0 -1  0]
 [ 0  0  0 -1  1]]
```

\*\*\*\*\*Vecindad de Neumann\*\*\*\*\*

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  0]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  0  0  0  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  0  0  5  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  0  6  5  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  7  6  5  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 8  7  6  5  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 8  7  6  5  4]
 [ 0  8 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 8  7  6  5  4]
 [ 9  8 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[ [ 8  7  6  5  4]
  [ 9  8 -1 -1  3]
  [ 0  9  0 -1  2]
  [ 0  0  0 -1  1]]
```

Ruta: ( 2 , 2 )

Ruta: ( 2 , 1 )

Ruta: ( 1 , 1 )

```
[ [ 8  7  6  5  4]
  [ 9  8 -1 -1  3]
  [10  9  0 -1  2]
  [ 0  0  0 -1  1]]
```

Ruta: ( 1 , 1 )

Ruta: ( 0 , 2 )

Ruta: ( 0 , 3 )

```
[ [ 8  7  6  5  4]
  [ 9  8 -1 -1  3]
  [10  9 10 -1  2]
  [ 0  0  0 -1  1]]
```

Ruta: ( 0 , 4 )

Ruta: ( 1 , 4 )

Ruta: ( 2 , 4 )

```
[ [ 8  7  6  5  4]
  [ 9  8 -1 -1  3]
  [10  9 10 -1  2]
  [ 0 10  0 -1  1]]
```

Ruta: ( 3 , 4 )

```
[ [ 8  7  6  5  4]
  [ 9  8 -1 -1  3]
  [10  9 10 -1  2]
  [11 10  0 -1  1]]
```

```
[ [ 8  7  6  5  4]
  [ 9  8 -1 -1  3]
  [10  9 10 -1  2]
  [11 10 11 -1  1]]
```

Los resultados observados anteriormente, son producto de la vecindad de **Neumann**, donde el usuario digita el número 1 y procede a introducir los valores que el algoritmo pide a medida que se va ejecutando. Las matrices que se observan, parten desde una posición final igual a 1 hasta una posición inicial que es hasta donde hará el recorrido. A medida que se van actualizando las posiciones, se va mostrando iterativamente los valores respectivos a cada posición.

Por último, se muestran los valores de una lista, en donde el primer valor hace referencia a una posición en X y el segundo en Y. Estos valores, son las rutas que toma el algoritmo del punto de inicio hasta llegar al punto de finalización. Toma la ruta más óptima.

1. John von Vecindades
2. Edward Forrest Moore
3. Salir

Elige una opcion

Introduce un numero entero: 2

```
[[ 0  0  5  4  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0  1  2]
 [ 0  0  0 -1  1]]
```

Fila posición inicial: 3

Columna posición inicial: 2

Fila posición final: 3

Columna posición final: 4

```
[[ 0  6  5  4  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

Matriz Original

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  0]
 [ 0  0  0 -1  0]
 [ 0  0  0 -1  0]]
```

```
[[ 0  6  5  4  4]
 [ 0  6 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

Matriz iniciada

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  0]
 [ 0  0  0 -1  0]
 [ 0  0  0 -1  1]]
```

```
[[ 7  6  5  4  4]
 [ 0  6 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

\*\*\*\*\*Vecindad de Moore\*\*\*\*\*

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  0]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 7  6  5  4  4]
 [ 7  6 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  0  0  0  0]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 7  6  5  4  4]
 [ 7  6 -1 -1  3]
 [ 0  7  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  0  0  0  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 7  6  5  4  4]
 [ 7  6 -1 -1  3]
 [ 0  7  7 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 0  0  0  4  4]
 [ 0  0 -1 -1  3]
 [ 0  0  0 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 7  6  5  4  4]
 [ 7  6 -1 -1  3]
 [ 7  7  7 -1  2]
 [ 0  0  0 -1  1]]
```

```
[[ 7  6  5  4  4]
 [ 7  6 -1 -1  3]
 [ 7  7  7 -1  2]
 [ 0  8  0 -1  1]]
```

```
[ [ 7  6  5  4  4]
  [ 7  6 -1 -1  3]
  [ 7  7  7 -1  2]
  [ 0  8  8 -1  1]]
```

```
[ [ 7  6  5  4  4]
  [ 7  6 -1 -1  3]
  [ 7  7  7 -1  2]
  [ 8  8  8 -1  1]]
```

Ruta: ( 2 , 2 )

Ruta: ( 1 , 1 )

Ruta: ( 0 , 2 )

Ruta: ( 0 , 3 )

Ruta: ( 1 , 4 )

Ruta: ( 2 , 4 )

Ruta: ( 3 , 4 )

```
1. John von Vecindades
2. Edward Forrest Moore
3. Salir
Elige una opcion
Introduce un numero entero: 3
```

Al finalizar la vecindad de **Neuman** y sus respectivas rutas, el programa retorna el menú y el usuario ingresa la opción número dos para iniciar la vecindad de **Moore**. El proceso es prácticamente igual que con el proceso de la vecindad de **Neuman**, la diferencia es que este tiene la propiedad de iterar la matriz no solo de forma recta (arriba, abajo, derecha e izquierda) este método también tiene la propiedad para poder moverse en sus diagonales.

Y al finalizar, se pueden notar que en las rutas el trayecto es mucho más corto que con la vecindad de **Neumann**, puesto que al iterar también en diagonales puede hacer su recorrido de inicio a final, mucho más rápido y en menos pasos.

Como último paso para completar la secuencia del programa. El usuario indica el índice número 3 en el menú de opciones para finalizar el programa