|  |  |
| --- | --- |
| Universidad del Valle de Guatemala | Jonathan Davis Alvarez Orozco -15842 |
| Algoritmos y Estructuras de Datos | Rodrigo Eduardo Barrios Estrada - 15009 |
| Sección 30 | Esteban Mario Francisco Hernández Molina- 15135 |

**Un robot sale de un laberinto**

Algoritmos para resolver un Laberinto:

* Algoritmo de Tremaux: creado por un Ingeniero Francés llamado Charles Tremaux en el siglo XIX. Se le considera el algoritmo más eficiente, ya que asegura la escapatoria de cualquier laberinto sin importar la forma. Para implementarlo, se debe ir marcando el camino que se está haciendo. Al llegar a un cruce se toma cualquier camino que no se haya recorrido antes, y si se llega a un punto sin salida se regresa al último cruce hecho. Si ya se hicieron todos los posibles caminos en el cruce, se regresa al anterior y así sucesivamente hasta llegar al final del laberinto.
* Regla de la mano derecha: Este algoritmo es bastante simple, consiste en colocar nuestra mano derecha apoyada en la pared, y seguir de esa manera hasta encontrar la salida. Es muy importante no despegar la mano de la pared ya que de esa manera se puede asegurar que se pasan por todas las paredes. A pesar de que es un algoritmo bastante sencillo posee aspectos negativos: la velocidad de escapatoria dependerá de la suerte a la hora de la colocación de la mano al iniciar.

Aunque ambos algoritmos nos pueden ayudar para salir de un laberinto espontáneamente, no todos son infalibles. Y esto es debido a que hasta el momento no existe algún algoritmo que sea tan eficiente y efectivo para resolver directamente un laberinto y conducirnos directamente a la salida.

Escogimos el algoritmo de Tremaux debido a que este nos pareció que es una forma más ordenada que nos conducirá a la solución del problema. Además, consideramos que la otra propuesta de la mano derecha es un algoritmo que tardaría mucho tiempo en ejecutarse; el robot tardaría mucho en encontrar la salida.

**Pseudocódigo del algoritmo de Tremaux**

Lista de contadores [ ]

Lista de rutas [ ]

Laberinto (dirección): //en dirección se ingresa izquierda o derecha, según la última tomada

Si (dirección=inicio): //cuando se comienza el laberinto

Agregar nuevo contador a la lista de contadores

Mientras (no hay pared enfrente):

sumar 1 al contador por cada paso

Si (hay pared enfrente):

Si (hay pared a la derecha):

Si (hay pared a la izquierda):

caminar atrás la cantidad de pasos del contador

Laberinto (adelante)

else:

voltear a la izquierda

Laberinto (izquierda)

else:

voltear a la derecha

Laberinto (derecha)

Si (dirección=izquierda):

Agregar nuevo contador a la lista

Mientras (no hay pared enfrente):

sumar 1 al contador por cada paso

Si (hay pared enfrente):

Si (hay pared a la derecha):

Si (hay pared a la izquierda):

caminar atrás la cantidad de pasos del contador

girar a la derecha

caminar atrás la cantidad de pasos del anterior contador

Laberinto (pop rutas)

else:

voltear a la izquierda

Laberinto (izquierda)

else:

voltear a la derecha

Laberinto (derecha)

Si (dirección=derecha):

Agregar nuevo contador a la lista

Mientras (no hay pared enfrente):

sumar 1 al contador por cada paso

Si (hay pared enfrente):

Si (hay pared a la derecha):

Si (hay pared a la izquierda):

caminar atrás la cantidad de pasos del contador

Laberinto (izquierda)

else:

voltear a la izquierda

Laberinto (izquierda)

else:

voltear a la derecha

Laberinto (derecha)

<http://www.matematicasdigitales.com/como-escapo-de-este-laberinto/>