

Universidad del Valle de Guatemala

Campus Central

Facultad de Ingeniería



Algoritmos y Estructura de Datos

PROYECTO ROBOT QUE SALE DE UN LABERINTO

Realizado por:

Gladys De La Roca 15755

Jackeline Hidalgo 15776

Andrea Pena 15127

Steven Rubio 15044

Catedrático: Lynett Garcia

Sección:

Guatemala, 3 de agosto del 2016

ALGORITMOS QUE EXISTEN PARA SALIR DE UN LABERINTO

De fuerza bruta

Es el algoritmo o técnica más simple, en general es una traducción directa del problema. Este consiste en probar todas las posibles posiciones del patrón en el texto. Requiere un espacio constante y realiza saltos de un carácter. Compara de izquierda a derecha y al analizar la complejidad en la realización de la búsqueda del patrón es de $O(mn)$ (URN 2008).

Lógica: Se sitúa el patrón en la primera posición y se compara caracter a caracter hasta encontrar un fallo o el final del patrón. Se pasa a la siguiente posición y se repite el proceso y se finaliza en el fin del texto (Sedgewick 1995).

Entonces este verifica para cada posición posible del texto en la que el patrón pueda concordar (Sedgewick 1995).

- Regla de la mano derecha: Es un algoritmo muy sencillo y ha sido utilizado desde hace cientos de años para encontrar salidas de laberintos, catacumbas o galerías subterráneas.

Este consiste en que hagamos uso de la mano derecha y la coloquemos sobre alguna pared y de esta manera recorrer el laberinto sin despegar la mano para asegurarnos que pasamos por todas las paredes. Eventualmente este nos llevara a la salida (Gámez 2015).

El tiempo que tome en salir del laberinto dependerá de la posición inicial de quien desee salir, quiere decir que el puede ser un método que tome bastante tiempo.

Recursivo

Es un método o algoritmo muy potente de diseño y razonamiento formal. Se relaciona naturalmente con la inducción y facilita la resolución de problemas. Este es utilizado cuando se sabe como resolver el problema de forma directa para cierto conjunto de datos y para el resto resolverlo utilizando la solución al mismo problema con datos "más simples" (Méndez 2002).

Este se basa en análisis de datos clasificándolos para determinar si se utilizara una solución directa y recursiva, es decir que para los casos directos podrá calcularse su solución de forma sencilla y para los recursivos a partir de los datos y resultados de los de caso directo solucionar de la misma manera para datos más pequeños o sencillos (Méndez 2002).

Para su implementación el algoritmos se utilizan acciones que puedan llamarse o invocarse como funciones o procedimientos. Se puede entender esto como copias del mismo código operadas sobre datos diferentes.

Tipos de recursiones (Méndez 2002)

- Simples: Una llamada recursiva en cada caso recursivo
 - Final: No requiere combinación de resultados
 - No final: Requiere combinación de resultados
- Múltiples: Más de una llamada recursiva en algún caso recursivo

Algoritmo de Backtracking

Algoritmo utilizado principalmente para resolver problemas en los cuales la solución consta de una serie de decisiones adecuadas a cierto objeto. El algoritmo genera todas las secuencias de una manera sistemática y organizada hasta encontrar la correcta y rápida de las soluciones (Díaz 2013)

Lógica: Esta trabaja con nodos que buscan encontrar todas las posibles soluciones a tomar, así generando un árbol con todas las posibles soluciones existentes (UGR 2011).

Soluciones: Determina las soluciones de los problemas buscando sistemáticamente en el espacio de soluciones, estas se representan en el árbol de soluciones asociadas al conjunto de soluciones del problema (UGR 2011).

Eficiencia: Las funciones de acotación se consideran eficientes ya que reducen considerablemente el número de nodos a explorar, sin embargo las funciones de acotación necesitan mucho tiempo para su correcta evaluación, por lo tanto se debe encontrar un equilibrio entre el tiempo de evaluación y reducción de nodos (UGR 2011).

Heurísticos

Se conoce como algoritmo Heurístico cuando la solución no es denominada de forma directa, sino mediante ensayos y pruebas. Este método consiste en generar varias posibles soluciones siguiendo un patrón establecido. Luego las posibles soluciones generan un criterio que los caracteriza, si una solución no es aceptada por esta característica se genera otra y los pasos dados con la solución anterior no se consideran. Este algoritmo también es denominado como backtracking ya que inherentemente da una vuelta atrás para generar una nueva solución (UTFSM, 2009)

Los algoritmos aproximados son un tipo de especial de algoritmo heurístico que siempre encuentra una solución y el error máximo de la solución obtenida está acotado por el valor máximo de la diferencia entre la solución obtenida y la óptima y el valor máximo de la razón entre la solución óptima y la obtenida (Alvarez, 2015).

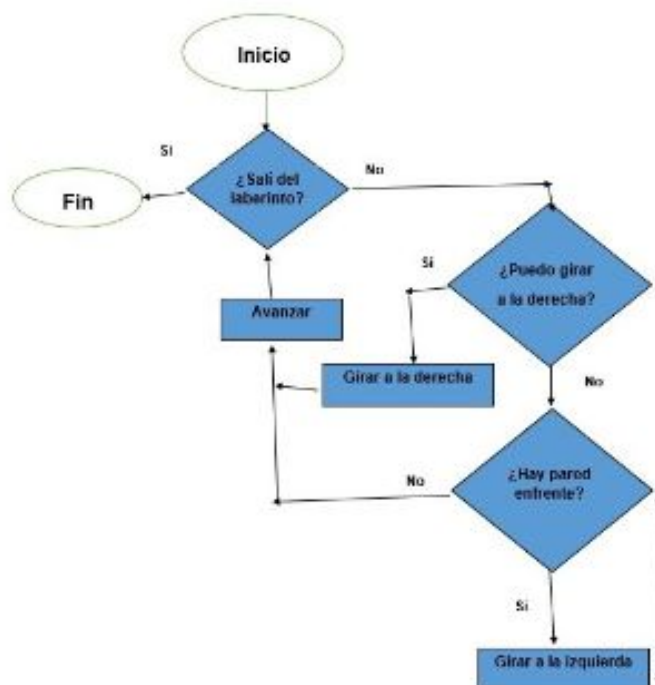
SELECCIÓN DEL ALGORITMO

El algoritmo seleccionado para resolver el laberinto es de tipo fuerza bruta, específicamente utilizando la regla de la mano derecha.

A pesar de la desventaja encontrada sobre el tiempo de demora en encontrar una salida, el cual depende de la posición inicial, podemos afirmar que es el más simple y funcional. Tomando en cuenta que la complejidad de un algoritmo de fuerza bruta es $O(mn)$.

Se implementa el de fuerza bruta ya que al tener la regla de la mano derecha se hace una traducción directa de este método para generar la solución. Consideramos que es el más funcional pues sea cual sea el laberinto sera seguro que al permanecer a su derecha cuando pueda, encontrará una alternativa que eventualmente lo llevará a la salida. Esto será de vital importancia, ya que se implementara el mismo algoritmo para resolver un laberinto físico y este debe poder resolverse en toda ocasión.

DIAGRAMA DE FLUJO



Bibliografía Citada

- Alvarez, L. 2015. Algoritmos Heurísticos y Aproximados. Consultado el 27 de julio de 2016. En: https://prezi.com/qd7kcxq_2ije/algoritmos-heuristicos-y-aproximados/
- Daz, A. 2013- Algoritmo de backtracking recursivo y no recursivo para la resolución de un laberinto y su aplicación en SDL. Consultado el 27 de julio de 2016. En: http://www.academia.edu/4607412/Algoritmo_de_Backtracking_Rekursivo_y_no_Rekursivo_para_la_Resoluci%C3%B3n_de_un_Laberinto_y_su_Aplicaci%C3%B3n_en_SDL
- Gámez, J. 2015. ¿Cómo escapo de este laberinto? Consultado el 25 de julio del 2016 En: <http://www.matematicasdigitales.com/como-escapo-de-este-laberinto/>
- Méndez, G. 2002. Diseño de algoritmos recursivos. Universidad Complutense de Madrid (UCM). Consultado el 25 de julio del 2016
En: http://www.fdi.ucm.es/profesor/gmendez/docs/edi0910/_02-Recursion.pdf
- Sedgewick, R. 1995. Algoritmos en C++. Addison-Wesley Publishing Company, Inc. Massachusetts, E.U.A. 750pp.
- UGR. 2011. Algoritmos Decsai. Consultado el 27 de julio de 2016- En: <http://elvex.ugr.es/decsai/algorithms/slides/5%20Backtracking.pdf>
- URN. 2008. Técnicas de diseño de Algoritmos. Consultado el 25 de julio del 2016
En: <http://www.fceia.unr.edu.ar/lcc/t312/archivos/08.ED.Dise%F1o.pdf>
- UTFSM, 2009. Programacion en Pascal, Capitulo 25. Algoritmos Heurísticos. Consultado el 27 de julio de 2016. En: <http://www2.elo.utfsm.cl/~lsb/pascal/clases/cap25.pdf>