CONTENIDO

MÓDULO II. ESTRATEGIA PARA LA SOLUCIÓN DE PROBLEMAS EN IA

UNIDAD I. SOLUCIÓN DE PROBLEMAS MEDIANTE BÚSQUEDA

Ilustración 1. *Solución de problemas mediante búsqueda.Pixabay.CCO*

En esta unidad se explicarán los aspectos generales sobre las técnicas de búsqueda en la Inteligencia Artificial: sus elementos y sus propiedades. Se describirán de forma breve las dos grandes clasificaciones de las técnicas de búsqueda y los diferentes tipos de algoritmos que los conforman, posteriormente se mostrarán ejemplos para una mejor comprensión de cada una de las técnicas que se estudiarán.

En esta primera parte correspondiente a la unidad 1 del módulo II se verán las técnicas de búsquedas sin información enfocándonos en la búsqueda en anchura y en profundidad. Dentro del campo de la Inteligencia Artificial, las técnicas de búsqueda son un tema fundamental. Debido a inherente complejidad de muchos problemas relacionados a la IA, las técnicas de búsqueda permiten representar el conocimiento y utilizar algoritmos para solucionarlos.

1.1. Aspectos generales sobre el diseño de programas de búsquedas

La búsqueda es un proceso que consiste en inspeccionar varias secuencias y escoger una de ellas para lograr un objetivo. Es decir, que se tiene una situación inicial cuya solución es el objetivo que se quiere lograr y se dispone de acciones simples que, al ejecutarlas en una secuencia en particular, pueden lograr o no el objetivo. Muchos de los problemas dentro de la IA se componen de ciertos elementos que son fundamentales para aplicar las técnicas de búsqueda, estos elementos son:

Ilustración 2. Elementos dentro de la IA. Freepik.CCBY

Los algoritmos de búsqueda tienen las siguientes propiedades:

- **Completitud:** El algoritmo devuelve una solución para cualquier entrada, si al menos una solución existe para esa entrada en particular.
- **Optimalidad:** Si la solución deducida por el algoritmo es a mejor solución.
- **Complejidad en tiempo:** Cuánto le toma al algoritmo completar su tarea.
- **Complejidad en espacio:** Espacio máximo de almacenamiento que el algoritmo requiere durante la búsqueda.

Las técnicas de búsqueda se clasifican en dos grandes grupos, de acuerdo con la cantidad de información disponible para realizar el proceso de búsqueda. La información puede ser sobre el espacio del problema o sólo sobre algunos estados. En la Ilustración 3 se muestra esta clasificación.

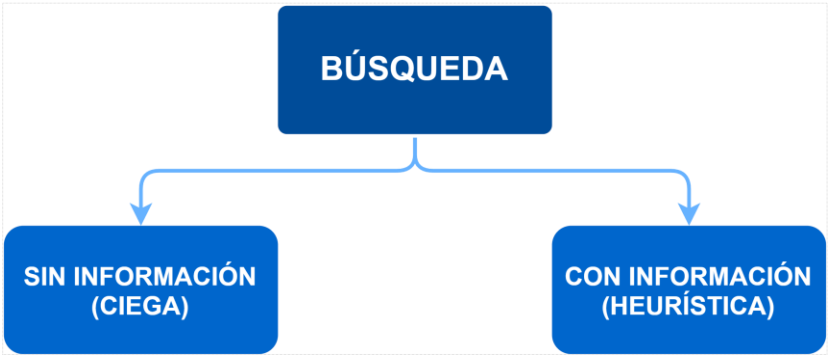

Ilustración 3. Clasificación de las técnicas de búsquedas. Autoria E. Cano

1.2 Búsqueda ciega o sin información

La búsqueda ciega consiste en una exploración sin información en el espacio de estados, es decir, no se tiene información sobre el dominio. Este tipo de búsqueda considera el camino más prometedor en el momento, no el camino óptimo para llegar al objetivo.

1.2.1 Búsqueda primero en anchura

La búsqueda primero en anchura se implementa utilizando a la estructura de datos denominada cola. El algoritmo empieza desde la raíz y atraviesa el árbol nivel por nivel, pasa por todos los nodos de un nivel antes de pasar a los nodos hijos, como se muestra en la Ilustración 3.

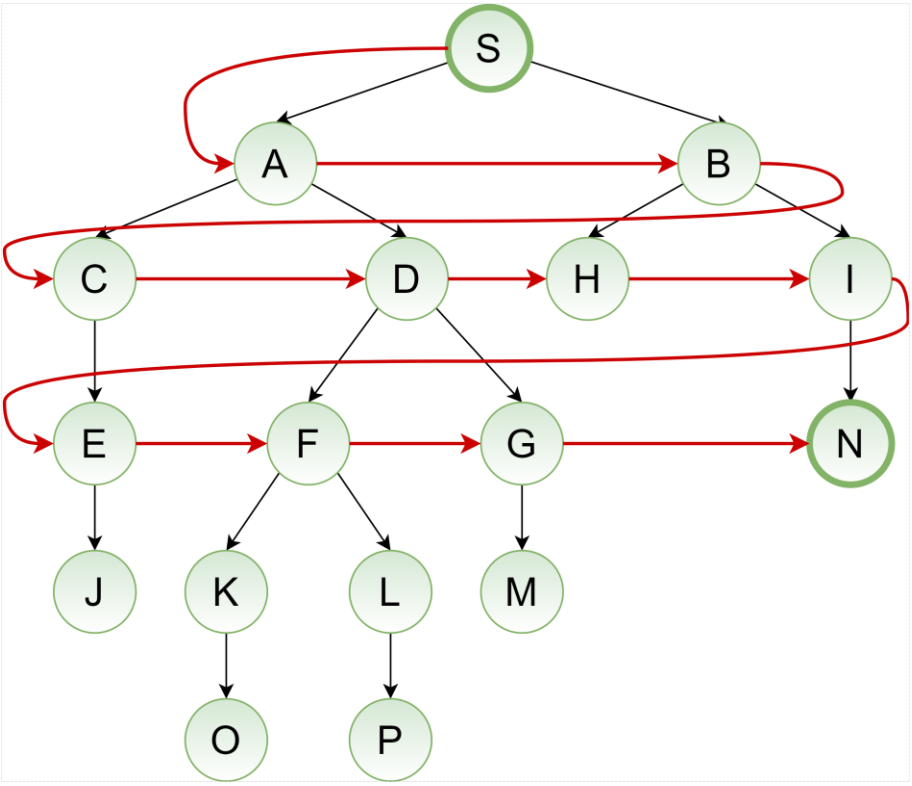

Ilustración 4. Búsqueda primero en anchura. Autoria E. Cano.

- **Completitud:** Siempre encuentra una solución, si existe.
- **Optimalidad:** Siempre encuentra el camino más corto, es decir, la solución de menor coste. Es más efectivo cuando todos los caminos hacia el nodo objetivo son de profundidad uniforme.
- **Complejidad en tiempo:** Exponencial. Requiere una considerable cantidad de tiempo para su ejecución.
- **Complejidad en espacio:** Exponencial. Requiere mucho espacio en la memoria.

Ejemplo: Realizar una búsqueda en anchura utilizando el siguiente árbol y los valores dados en el nodo inicial y el nodo objetivo.

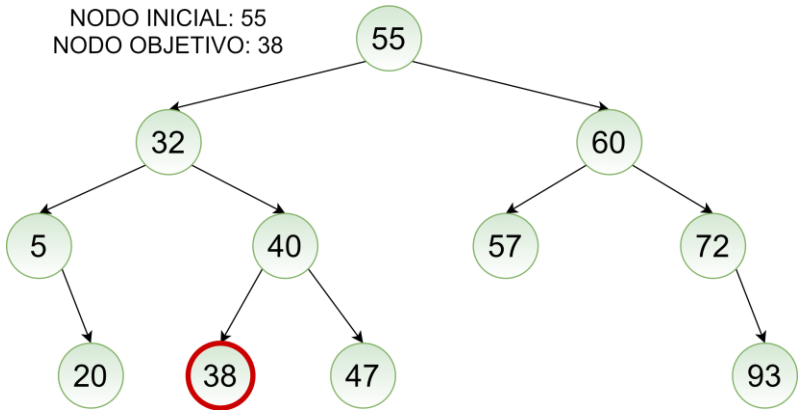

Ilustración 5. Búsqueda primero en anchura. Autoria E. Cano

Solución:

- **Paso 1**

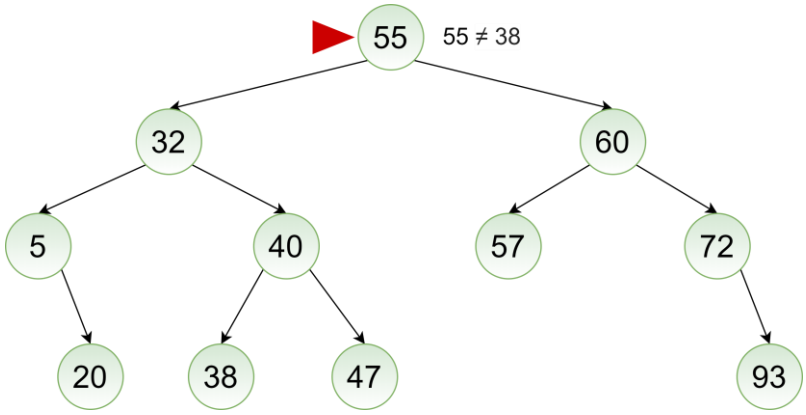

Ilustración 6. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 2**

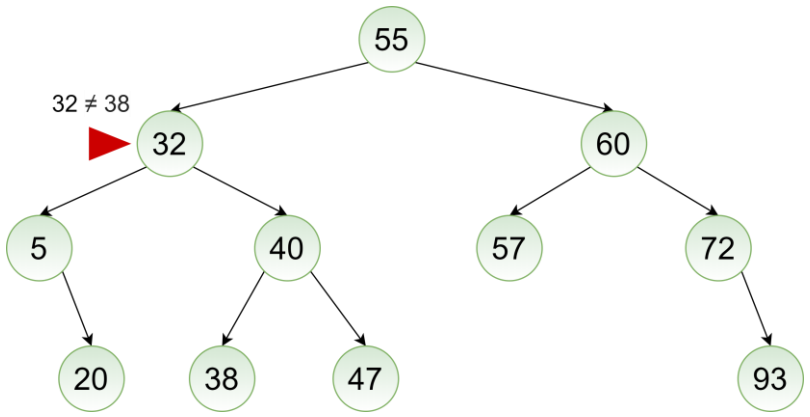

Ilustración 7. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 3**

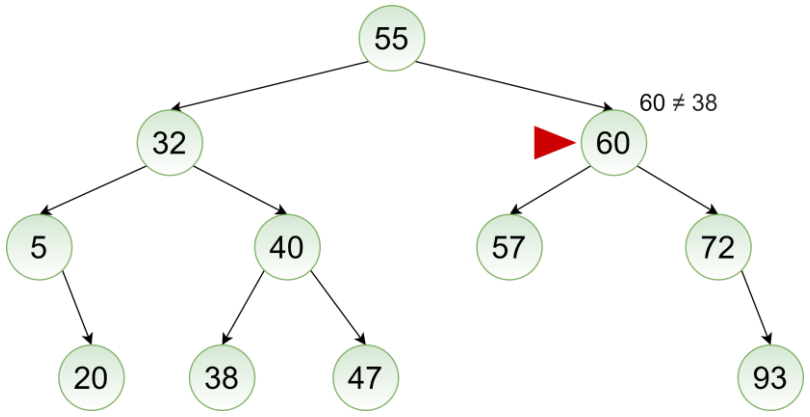

Ilustración 8. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 4**

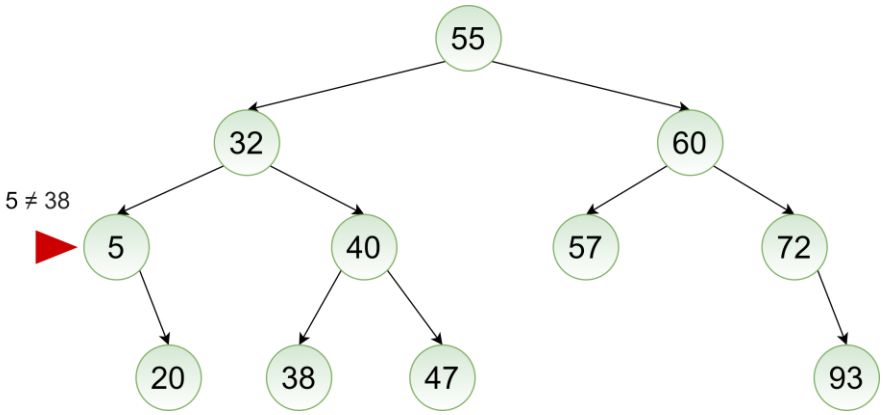

Ilustración 9. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 5**

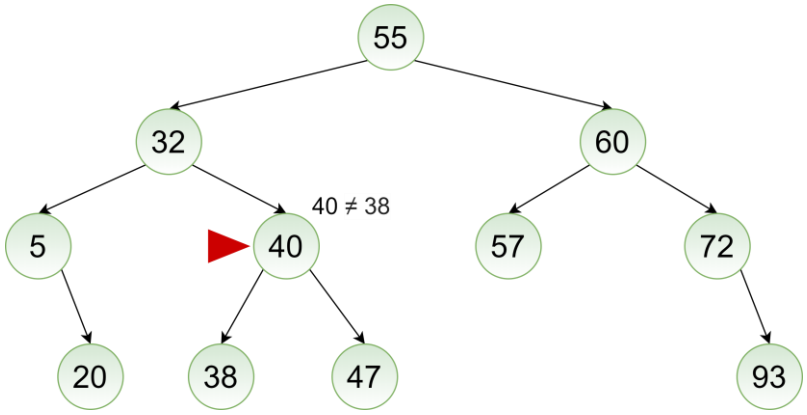

Ilustración 10. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 6**

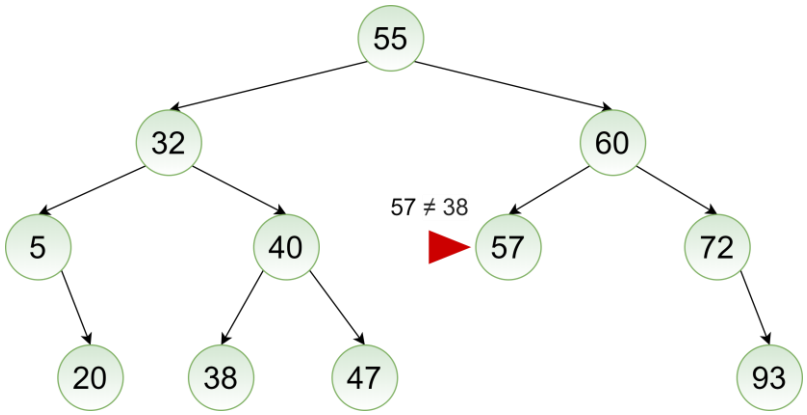

Ilustración 11. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 7**

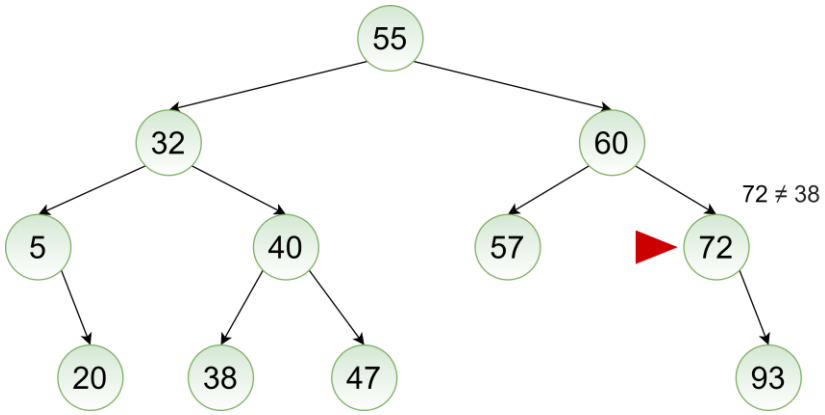

Ilustración 12. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 8**

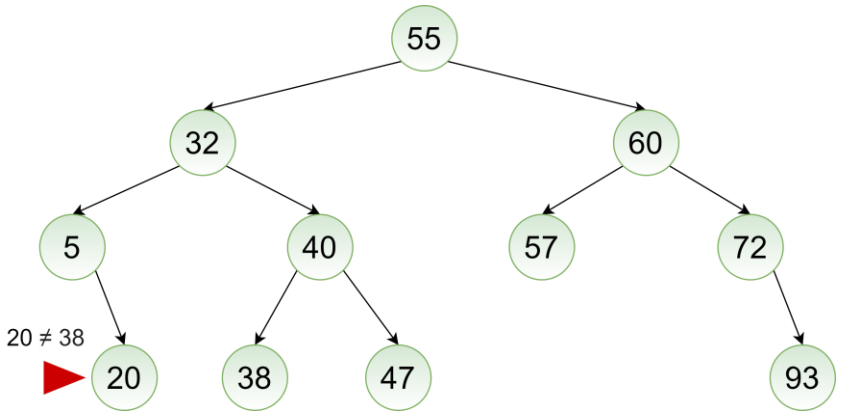

Ilustración 13. Búsqueda primero en anchura. Autoria E. Cano

- **Paso 9**

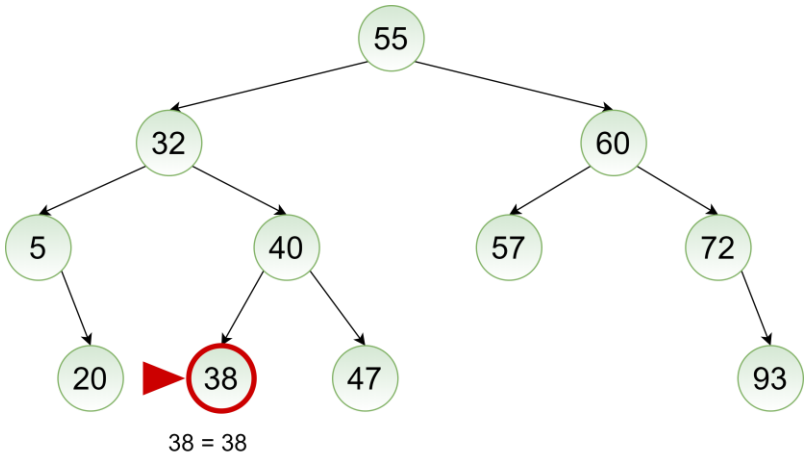

Ilustración 14. Búsqueda primero en anchura. Autoria E. Cano

1.2.2 Búsqueda primero en profundidad

La búsqueda primero en profundidad se implementa utilizando a la estructura de datos denominada pila. El algoritmo empieza desde la raíz y explora a todos los nodos de una sola rama del árbol. Si no encuentra una solución o termina la búsqueda por esa rama, entonces hace una vuelta hacia atrás (denominado backtracking) y sigue por otra rama. Sus principales desventajas son que la búsqueda puede no finalizar si se trata de un árbol infinito y puede presentarse bucles infinitos. En la Ilustración 4 se muestra cómo se comporta la búsqueda en profundidad.

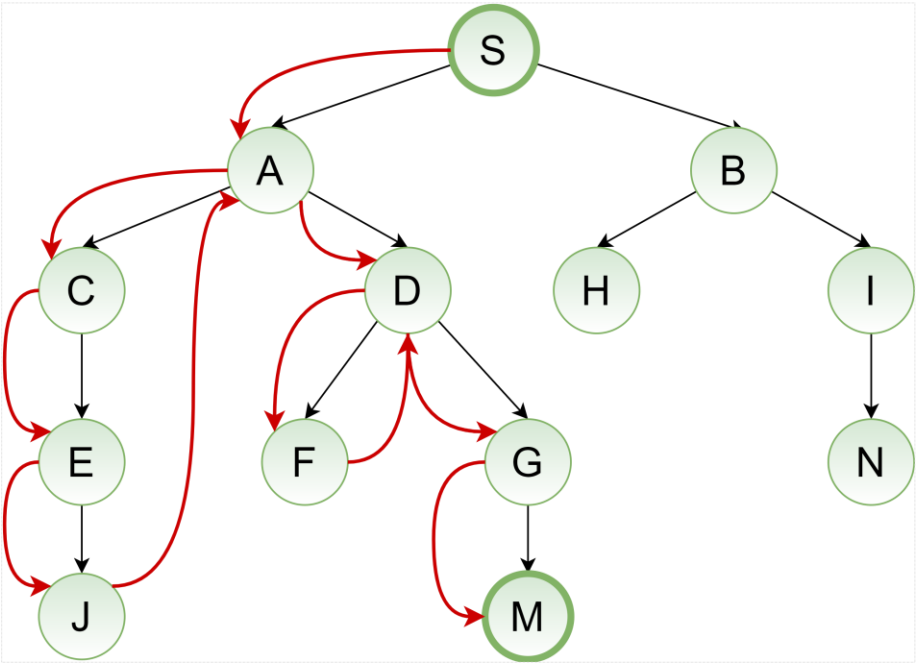

Ilustración 15. Búsqueda primero en profundidad. Autoria E. Cano

- **Completitud:** Siempre encuentra una solución, si existe y el árbol es finito.
- **Optimalidad:** No garantiza la solución de menor coste.
- **Complejidad en tiempo:** Exponencial. Genera los mismos conjuntos de nodos que en la búsqueda primero por anchura, pero en un orden diferente.
- **Complejidad en espacio:** Lineal. Requiere menos espacio en la memoria porque el algoritmo sólo necesita almacenar una pila con los nodos del camino desde la raíz hasta el nodo actual.

Ejemplo: Realizar una búsqueda en profundidad utilizando el siguiente árbol y los valores dados en el nodo inicial y el nodo objetivo.

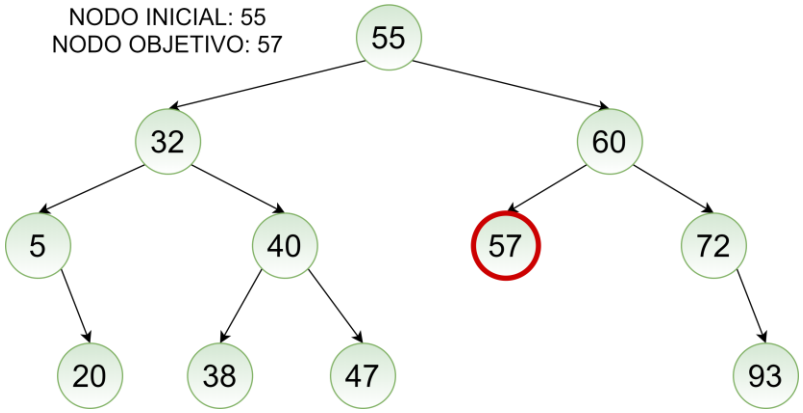

Ilustración 16. Búsqueda primero en profundidad. Autoria E. Cano

Solución:

- **Paso 1**

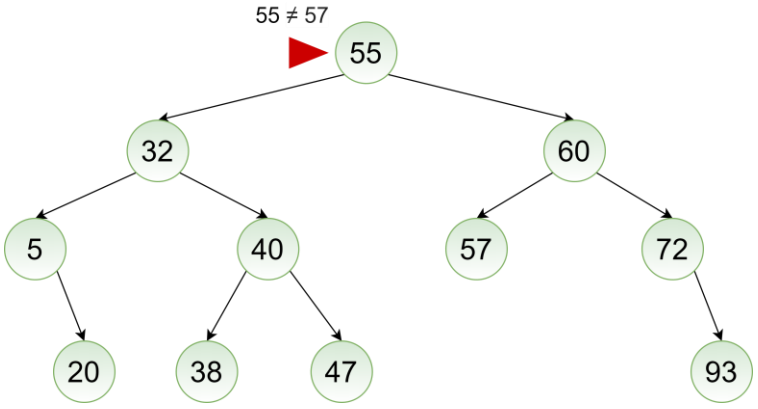

Ilustración 16. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 2**

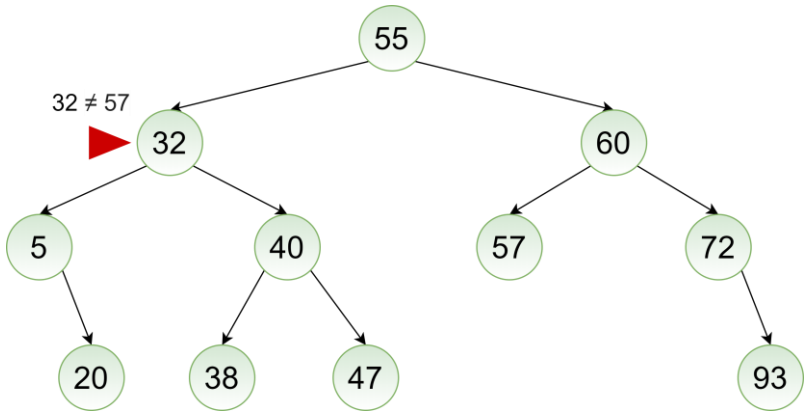

Ilustración 17. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 3**

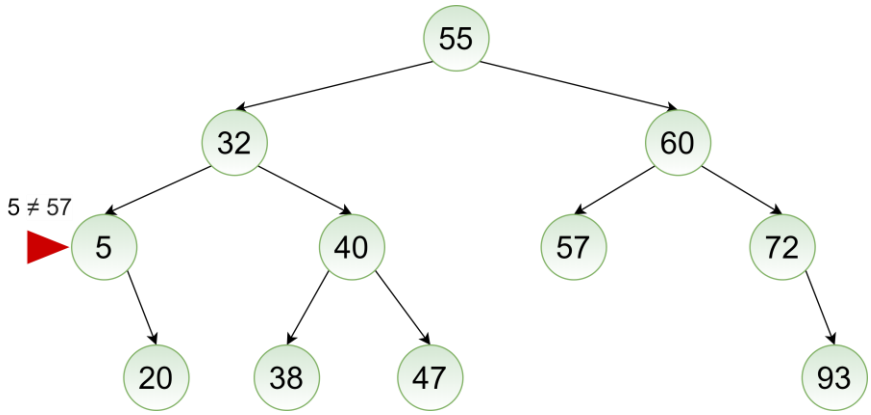

Ilustración 18. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 4**

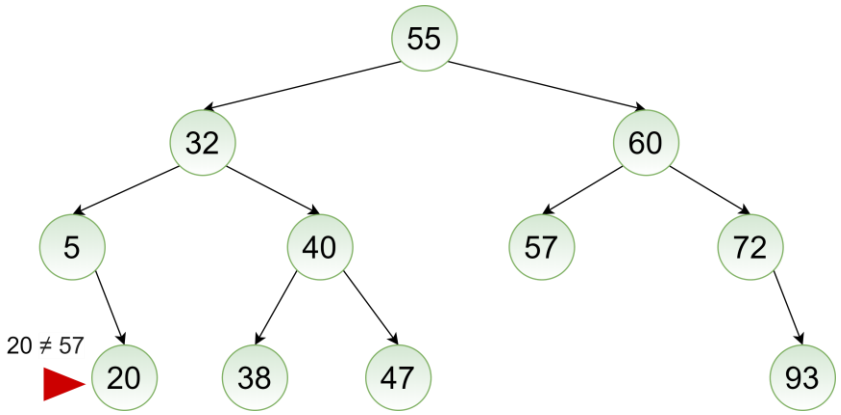

Ilustración 19. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 5**

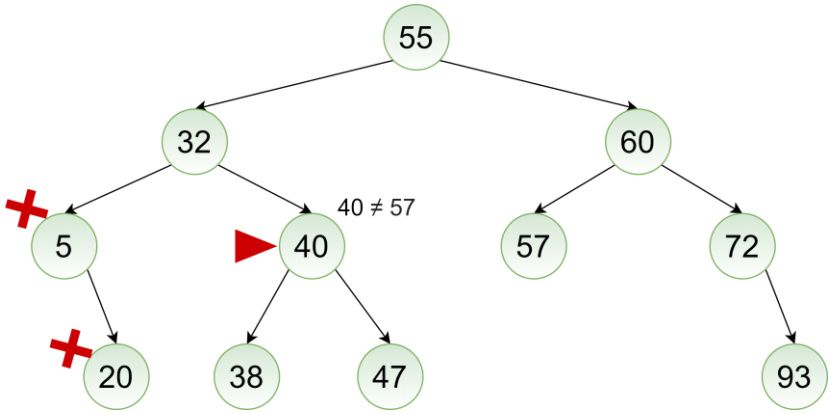

Ilustración 20. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 6**

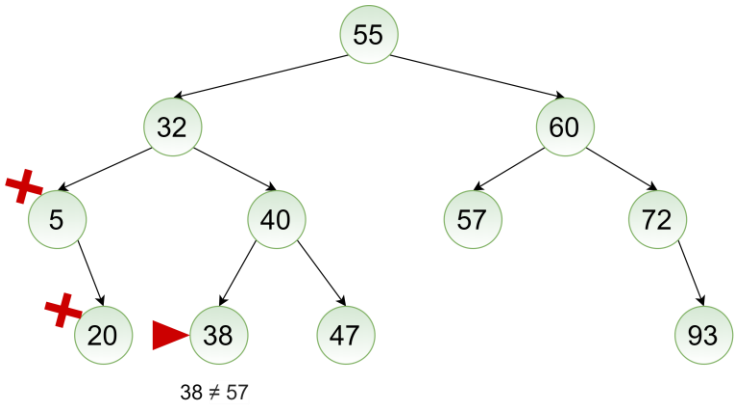

Ilustración 21. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 7**

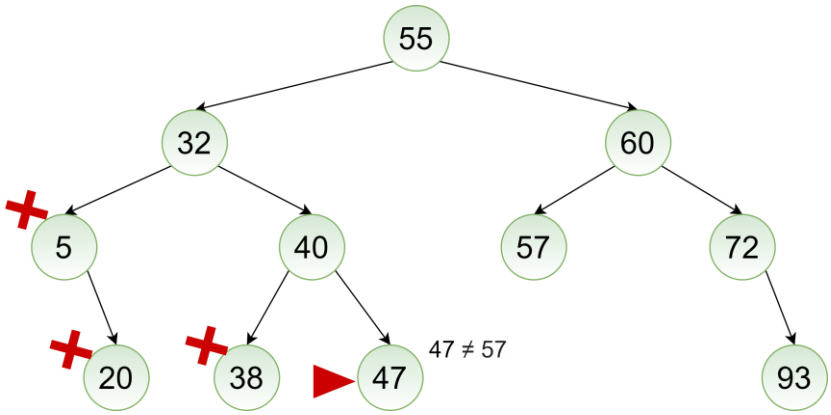

Ilustración 22. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 8**

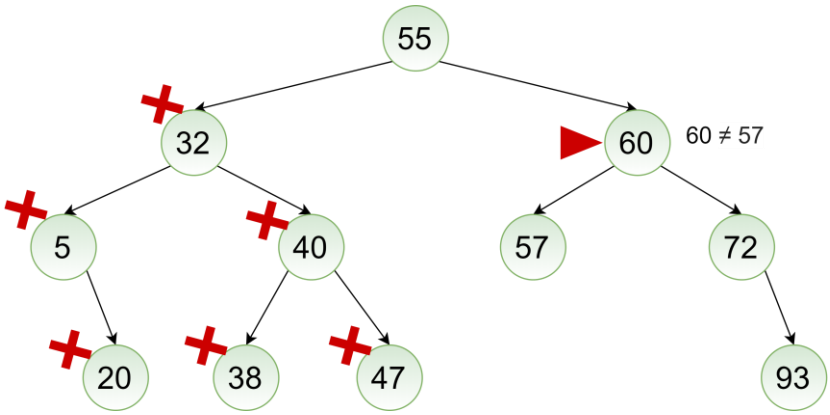

Ilustración 21. Búsqueda primero en profundidad. Autoria E. Cano

- **Paso 9**

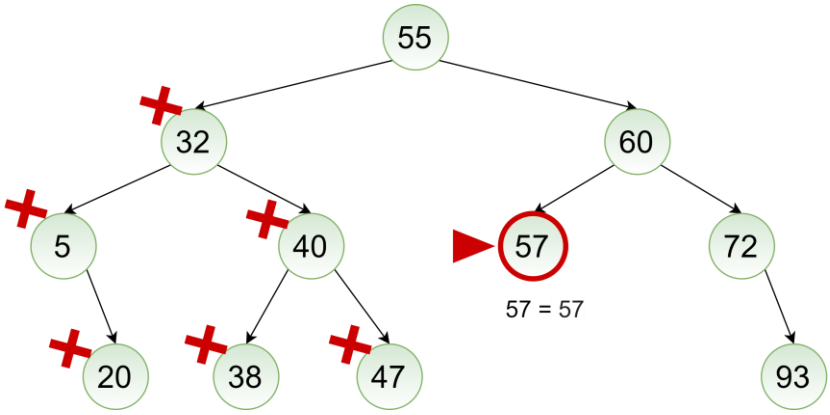

Ilustración 22. Búsqueda primero en profundidad. Autoria E. Cano

- ***Búsqueda en profundidad iterativa***

La búsqueda en profundidad iterativa combina las mejores características de la búsqueda en anchura y la búsqueda en profundidad.

En este tipo de búsqueda, el algoritmo funciona de la siguiente manera:

1. Se establece una profundidad predefinida.

2. El árbol se desarrolla realizando la búsqueda en profundidad hasta el límite predefinido en el paso 1.

3. Si encuentra una solución el proceso de búsqueda finaliza. De lo contrario se predefine una nueva profundidad y se repiten los pasos.

La búsqueda en profundidad iterativa presenta las siguientes características:

- **Completitud:** Siempre encuentra una solución, si existe, sin importar si el árbol es finito o no.
- **Optimalidad:** Garantiza la solución de menor coste.

En la Ilustración 23 se muestra cómo se aplica la búsqueda en profundidad iterativa:

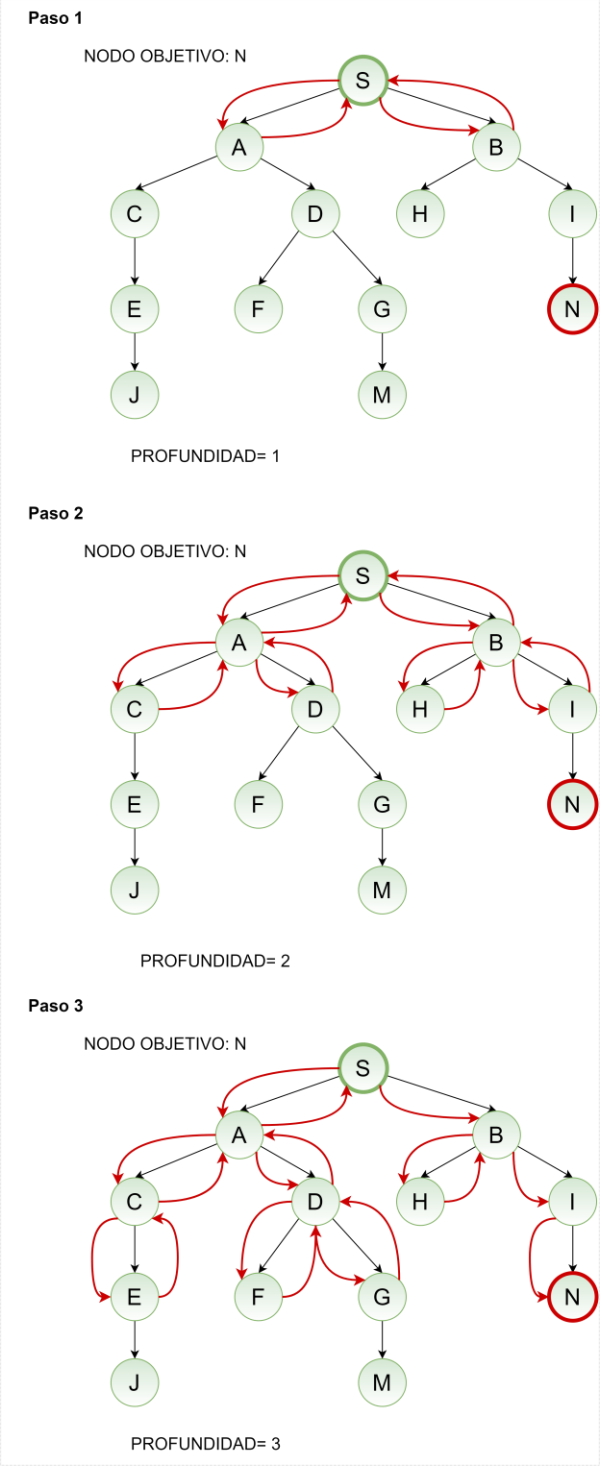

Ilustración 23. Búsqueda en profundidad iterativa. Autoria E. Cano

Ejemplo: Realizar una búsqueda en profundidad iterativa utilizando el siguiente árbol. El valor a buscar es: 57.

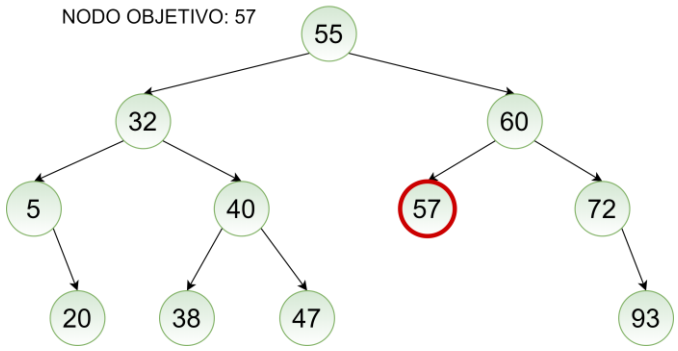

Ilustración 24. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 1**

Se predefine la profundidad como PROFUNDIDAD= 1

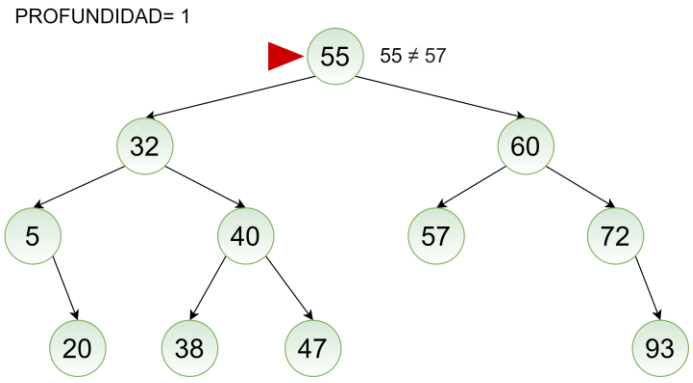

Ilustración 25. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 2**

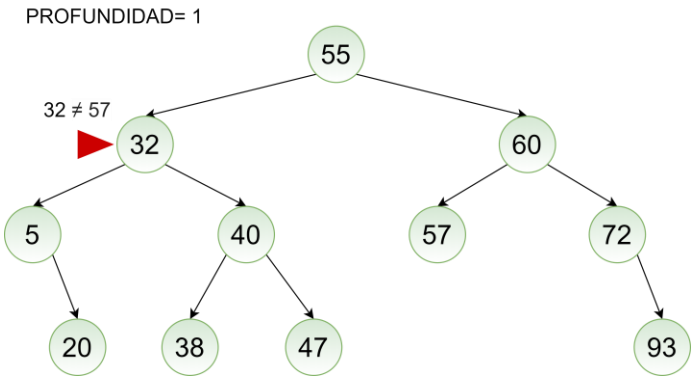

Ilustración 26. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 3**

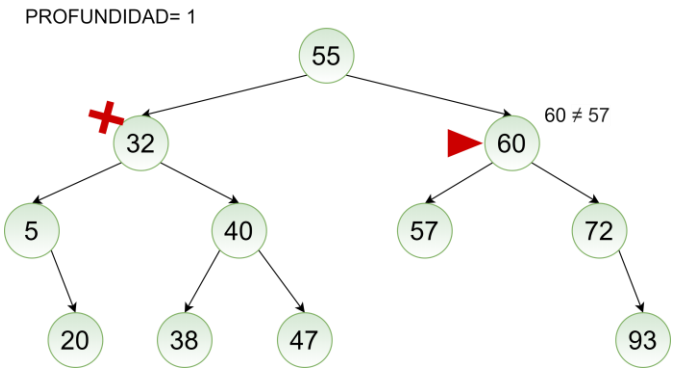

Ilustración 27. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 4**

Como no se encontró la solución, ahora se predefine la profundidad como PROFUNDIDAD= 2

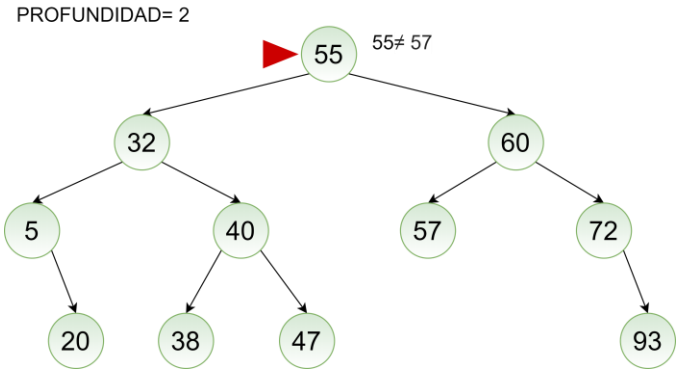

Ilustración 28. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 5**

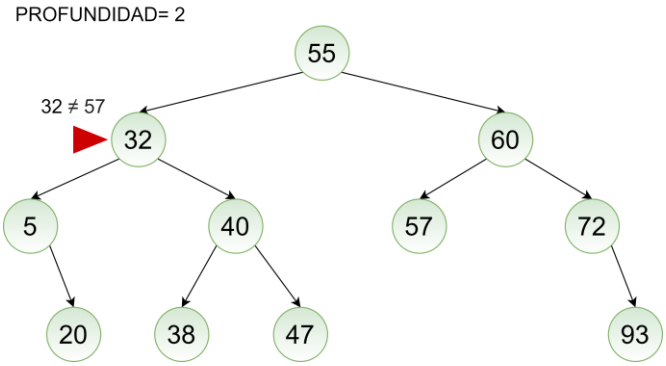

Ilustración 29. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 6**

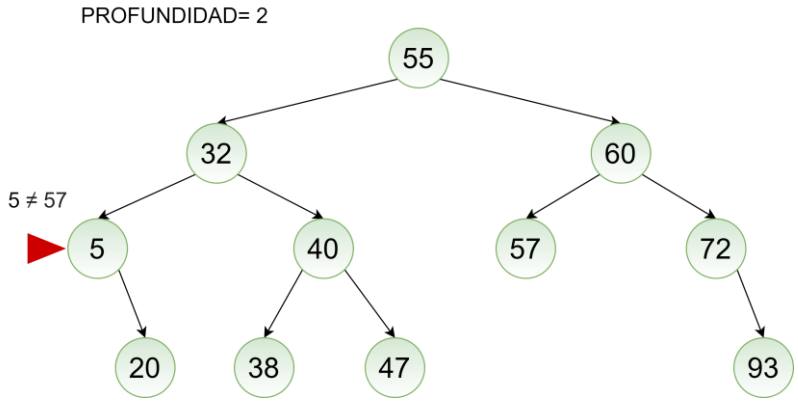

Ilustración 30. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 7**

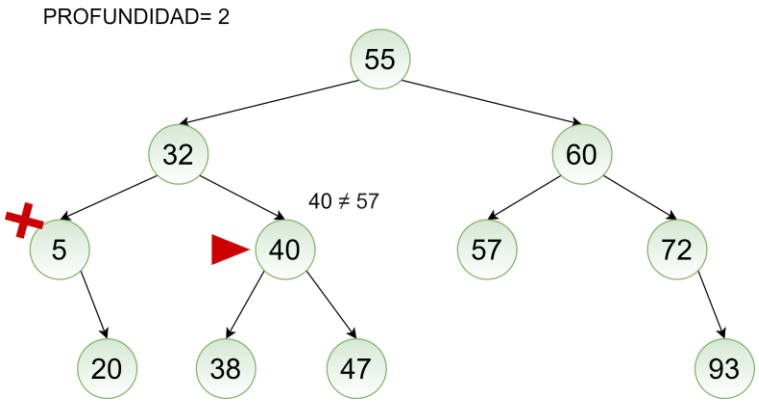

Ilustración 31. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 8**

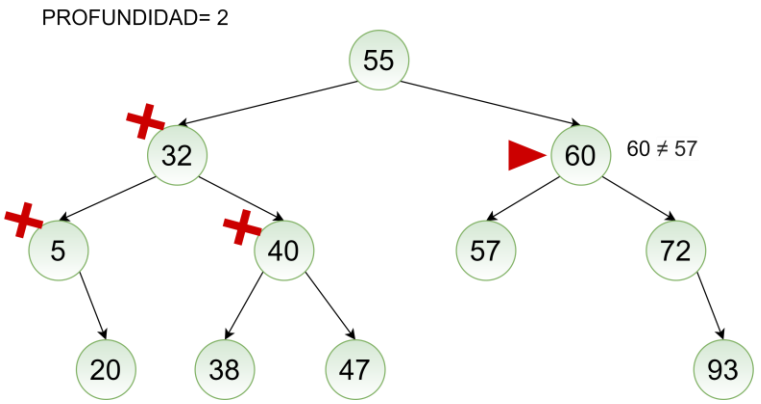

Ilustración 24. Búsqueda en profundidad iterativa. Autoria E. Cano

- **Paso 9**

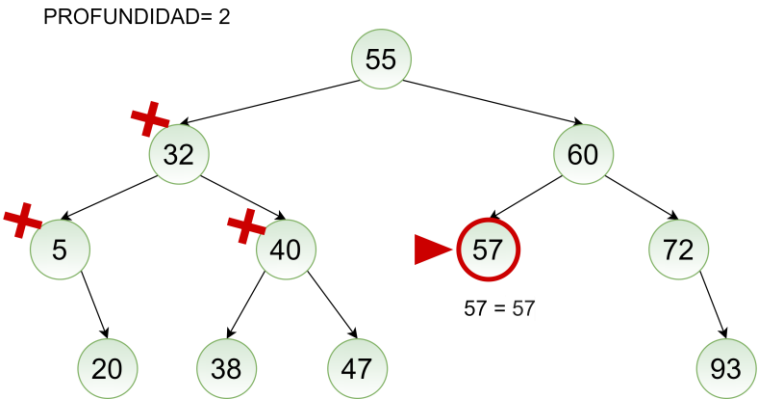

Ilustración 25. Búsqueda en profundidad iterativa. Autoria E. Cano