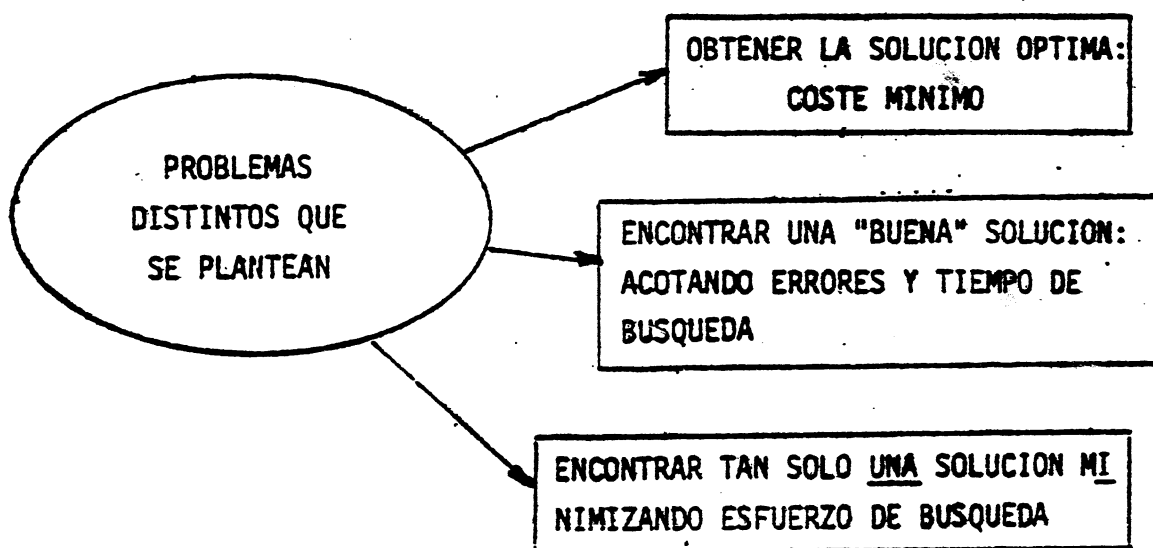
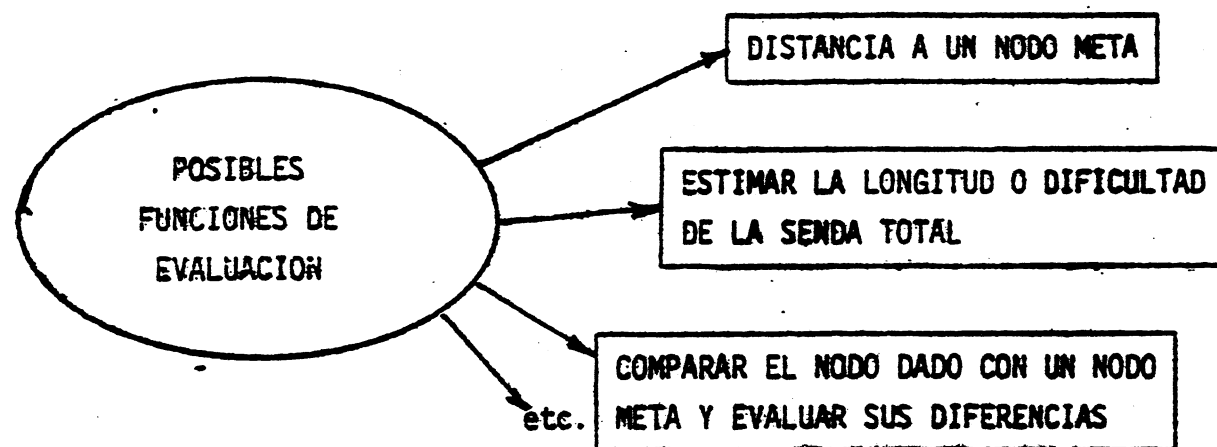


I. METODOS DE SOLUCION EN EL ESPACIO DE ESTADO

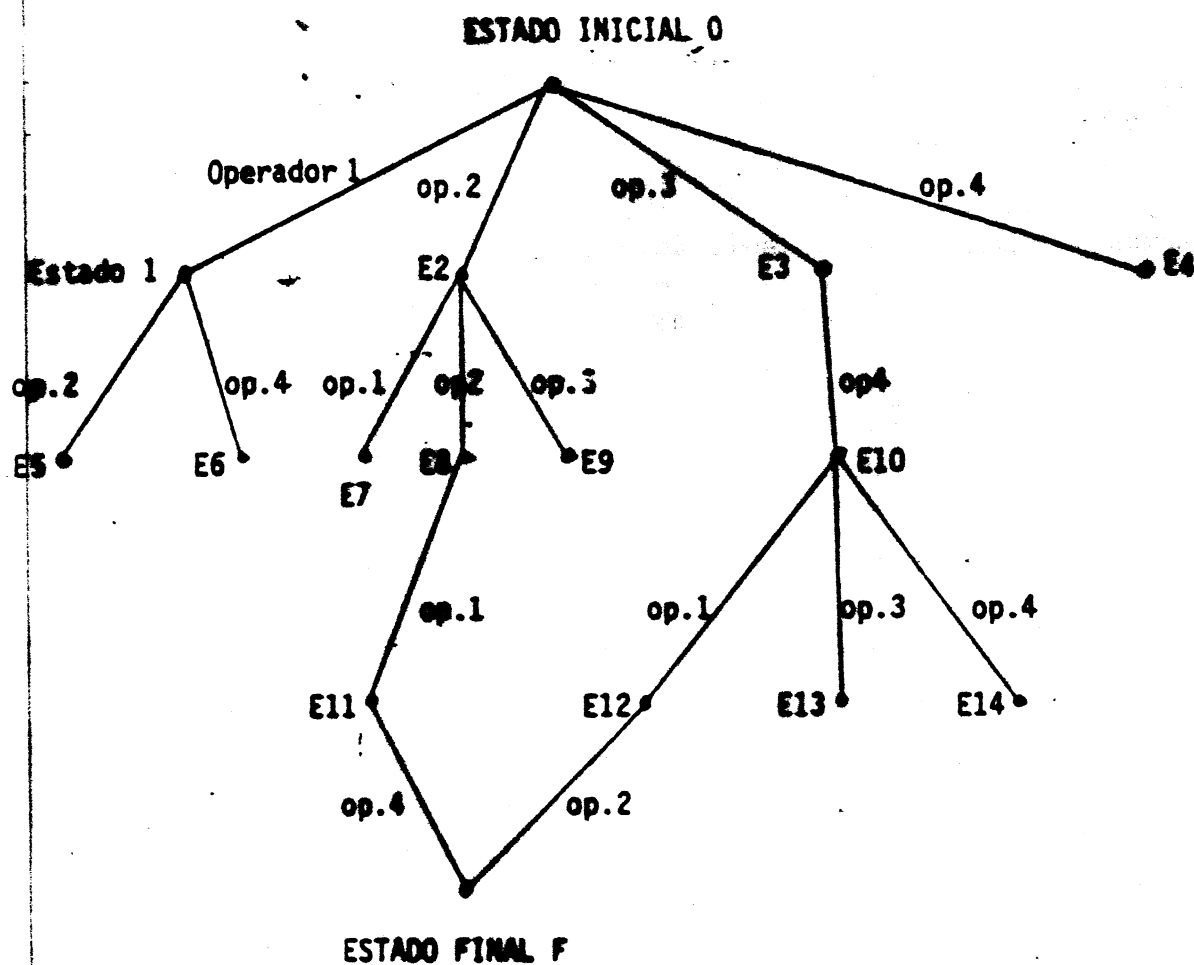
2. BUSQUEDA HEURISTICA

2-A. BUSQUEDA ORDENADA



I. REPRESENTACION EN EL ESPACIO DE ESTADO

- ESQUEMA GEOMETRICO DE LA SOLUCION



LA SOLUCION PUEDE SER MULTIPLE:

0 - 2 - 8 - 11 - F; y ; 0 - 3 - 10 - 12 - F

Y DE NIVELES DISTINTOS

METODOS DE FUERZA BRUTA

Los algoritmos de búsqueda más generales son los de fuerza bruta, puesto que no requieren ningún tipo de conocimiento específico del dominio. Sólo requieren una descripción de estados, un conjunto de operadores legales, el estado inicial y la descripción del estado meta. Los métodos más importantes de búsqueda dentro de esta categoría son:

- Búsqueda a lo ancho
- Búsqueda en profundidad
- Búsqueda bidireccional

BUSQUEDA A LO ANCHO

La búsqueda a lo ancho empieza por generar todos los sucesores del nodo raíz. Esto se llama **EXPANDIR** un nodo. Después se expanden todos los nodos sucesores, generándose todos los nodos del segundo nivel del árbol. Este método continúa expandiendo nivel a nivel hasta encontrar la meta.

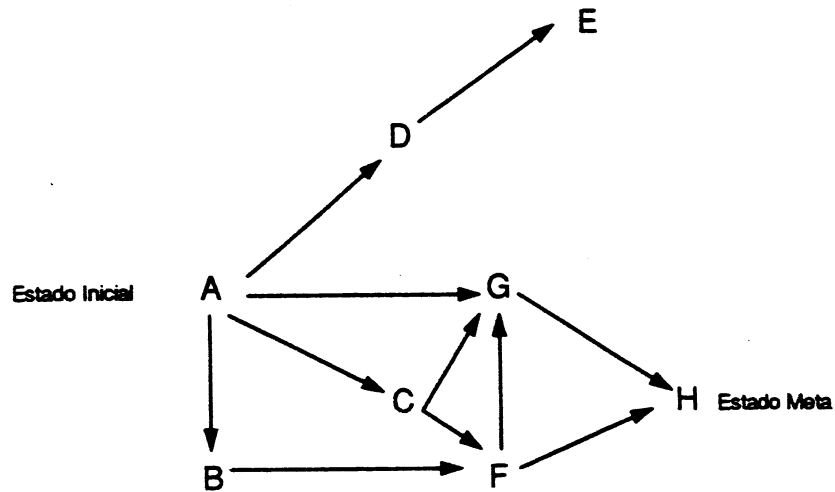
Puesto que la búsqueda a lo ancho nunca explora un nodo del árbol hasta que todos los nodos de menor nivel han sido generados y explorados, una vez que se encuentra un camino a la meta éste será el de menor longitud. Este método de búsqueda siempre encuentra una solución óptima.

ALGORITMO DE BUSQUEDA A LO ANCHO

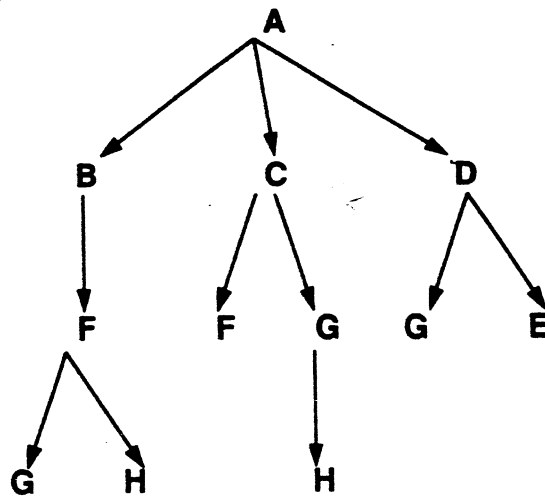
- 1.- Construir una cola de un elemento, elemento raíz.
- 2.- Hasta que la cola esté vacía o se haya llegado a la meta, determinar si el primer elemento de la cola es el nodo meta.
 - 2.a.- Si el elemento es el nodo meta, entonces no hacer nada.
 - 2.b.- Si el elemento no es el nodo meta, entonces retirar el primer elemento de la cola y añadir los hijos de este primer elemento (si los tiene) a la cola (por detrás de los demás elementos).
- 3.- Si se ha encontrado el nodo meta anunciar éxito y en caso contrario fracaso.

Ejemplo:

Supongamos que tenemos el siguiente mapa de carreteras y que estamos situados en el punto A y queremos llegar a H.



Solución:



I. METODOS DE SOLUCION EN EL ESPACIO DE ESTADO

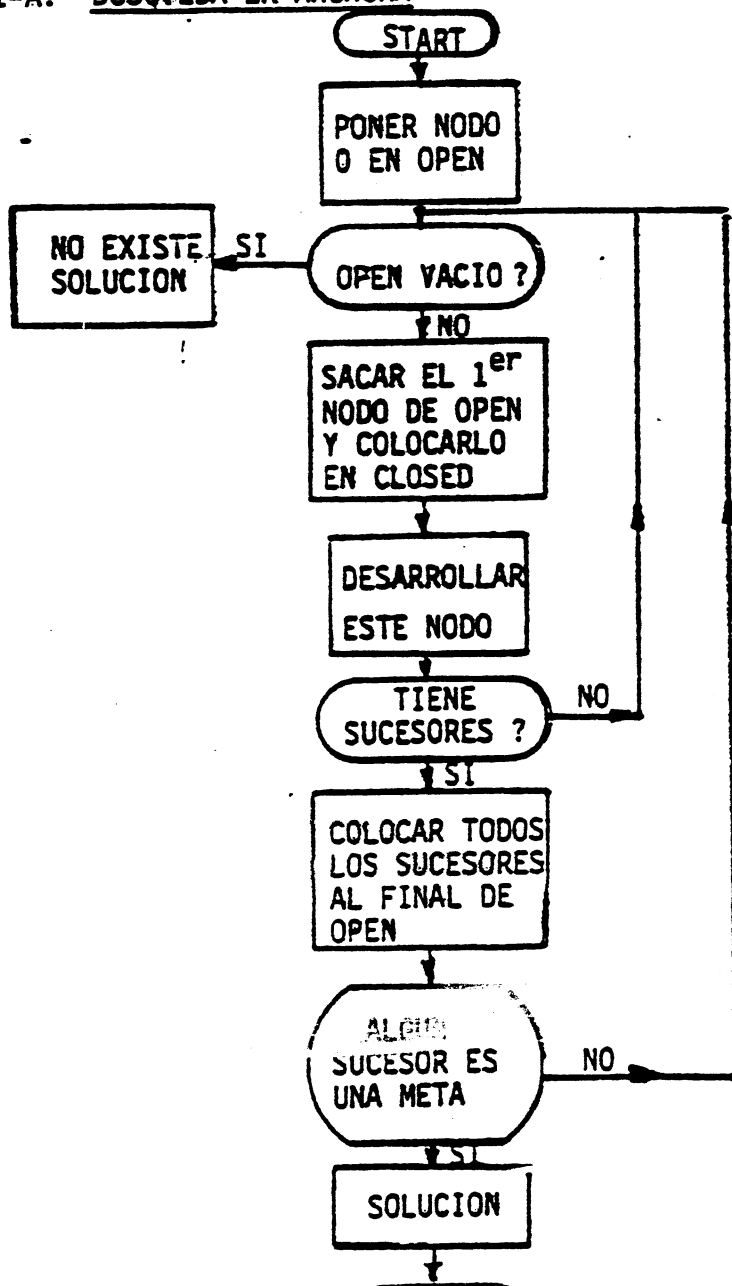
1. BUSQUEDA A CIEGAS

- EXPLOSION COMBINATORIA
- DEBE RECORRER TODO EL GRAFO
- NO TIENE DEMASIADO INTERES

HIPOTESIS USUALES

- EL GRAFO DEL ESPACIO DE ESTADO ES ARBOL
- A CADA NODO HIJO SE LE ASOCIA UN INDICATIVO DEL NODO-PADRE.

1-A. BUSQUEDA EN ANCHURA

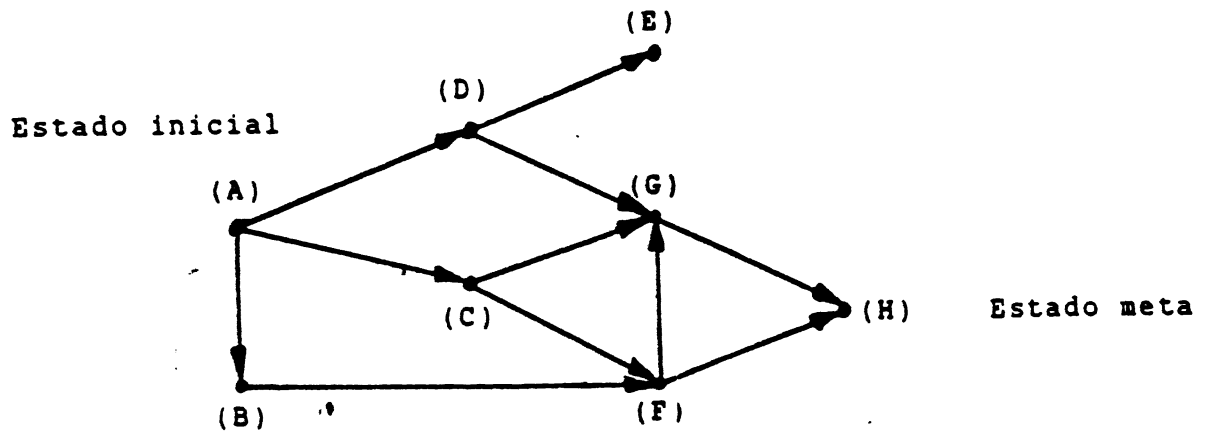


ALGORITMO DE BUSQUEDA A LO ANCHO

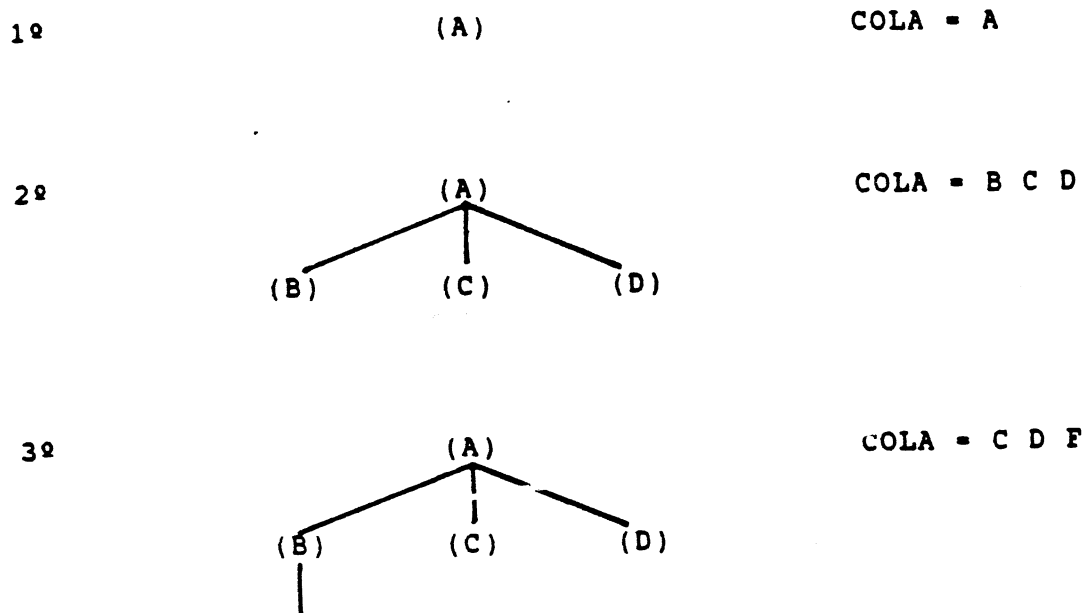
- 1.- Construir una cola de un elemento, elemento raíz.
- 2.- Hasta que la cola esté vacía o se haya llegado a la meta, determinar si el primer elemento de la cola es el nodo meta.
 - 2.a.- Si el elemento es el nodo meta, entonces no hacer nada.
 - 2.b.- Si el elemento no es el nodo meta, entonces retirar el primer elemento de la cola y añadir los hijos de este primer elemento (si los tiene) a la cola (por detrás de los demás elementos).
- 3.- Si se ha encontrado el nodo meta anunciar éxito y en caso contrario fracaso.

EJEMPLO

Supongamos que tenemos el siguiente mapa de carreteras y que estamos situados en el punto A y que queremos llegar a H.

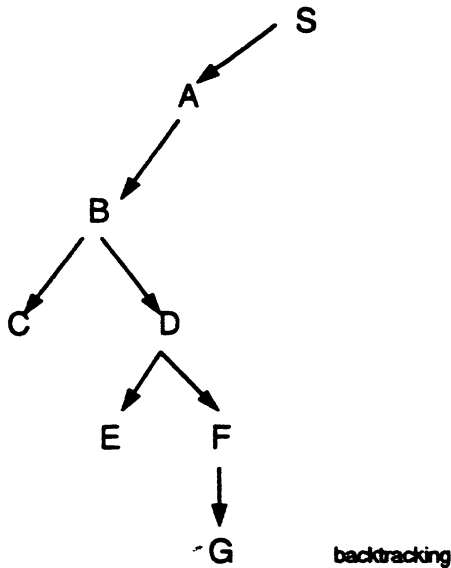


SOLUCION



BUSQUEDA EN PROFUNDIDAD.

Un método de búsqueda que remedia las limitaciones de espacio planteadas por la búsqueda a lo ancho, es el de profundidad. El método de búsqueda en profundidad, primero, genera un sucesor del nodo raíz, para luego generar uno de sus sucesores y continuar extendiendo este único camino hasta que termina naturalmente, o se recorte (**poda**) en un nivel dado. En cuyo caso retrocede y genera otro sucesor del último nodo generado en el camino en curso hasta que ese camino nuevo alcance su fin etc....



Por lo general **PROFUNDIDAD** extiende sus nodos de una manera **LIFO**. La búsqueda en profundidad genera hasta una profundidad "**d**" el mismo conjunto de nodos que la búsqueda a lo ancho (en diferente orden) por lo tanto la complejidad del tiempo es también una función exponencial $O(b^d)$.

Su ventaja, sin embargo, está en que es más eficiente en términos de memoria (espacio), puesto que **profundidad** sólo tiene que memorizar el camino en curso y la longitud máxima de este camino es "**d**" nodos, la complejidad espacial es $O(d)$ que es una función lineal de "**d**".

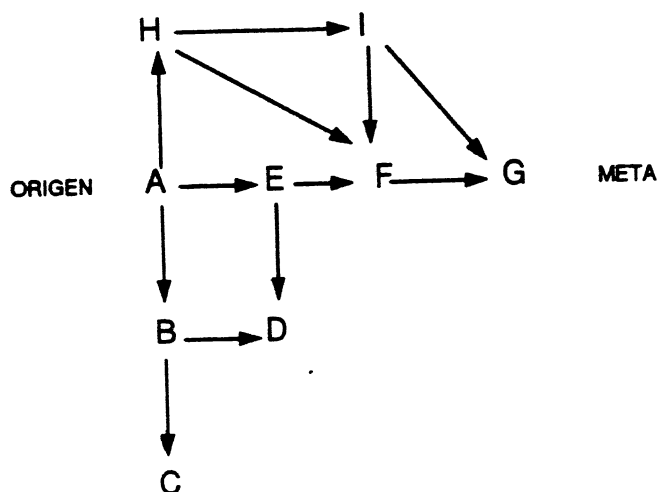
La desventaja está en que si la solución buscada no es uno de los nodos de un camino de la izquierda (a una profundidad prudencial) este método requiere un punto de corte (**poda**). Sin este corte o poda, la profundidad de este camino podría ser casi infinita al poder en algún caso meternos en un bucle.

Aunque el punto de corte ideal sería la "profundidad solución" (**d**), este valor casi nunca se conoce de antemano, y si la profundidad de corte la hacemos **c** (menor que **d**), el algoritmo terminará sin encontrar la solución. Además, si $c > d$ el precio que se paga también es grande ya que el tiempo de ejecución se pone en $O(b^c)$ y la primera solución que se encuentre puede que no sea la óptima en tiempo.

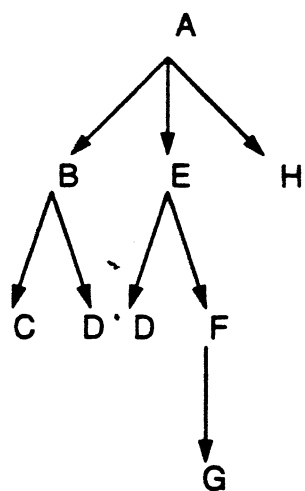
ALGORITMO DE BUSQUEDA EN PROFUNDIDAD

- 1.- Crear una **PILA** de un sólo elemento, el elemento raíz.
- 2.- Hasta que la pila esté vacía, o el elemento meta se haya alcanzado; determinar si el primer elemento de la pila es el nodo meta.
 - 2.a.- Si el primer elemento es el nodo meta, entonces no hacer nada.
 - 2.b.- Si el primer elemento no es el nodo meta, entonces retirar el primer elemento de la pila y añadir (si los hay) los hijos de este elemento a la pila (por delante de todos los demás).
- 3.- Si se ha encontrado el nodo meta anunciar éxito y en caso contrario fracaso.

EJEMPLO



SOLUCION



Meta

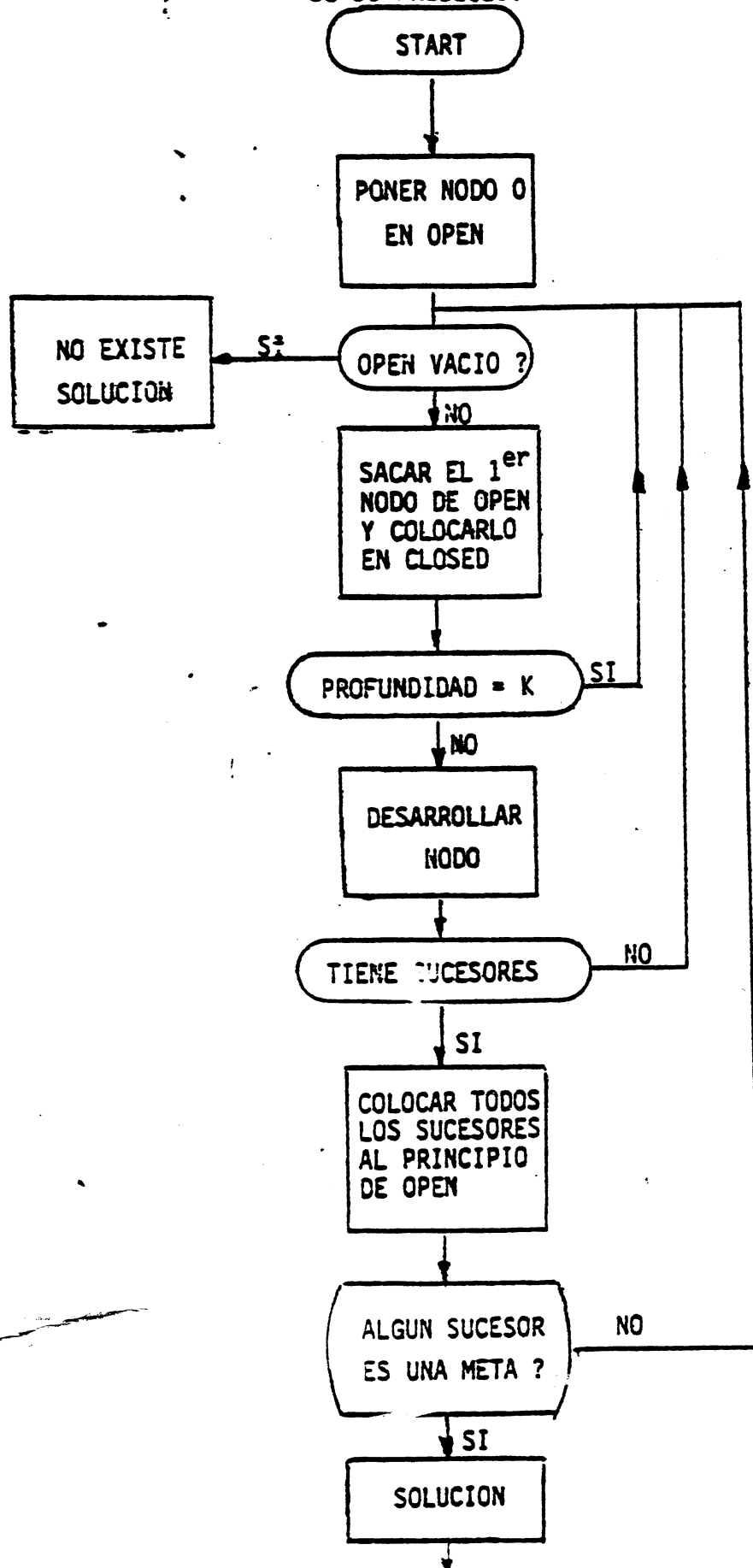
1. BUSQUEDA A CIEGAS

I-18.

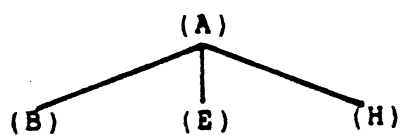
1-B. BUSQUEDA EN PROFUNDIDAD (LIMITADA)

PROFUNDIDAD: - LA PROFUNDIDAD DEL NODO INICIAL ES 0.

- LA PROFUNDIDAD DE CUALQUIER NODO ES 1 + LA PROFUNDIDAD DE SU PREDECESOR.

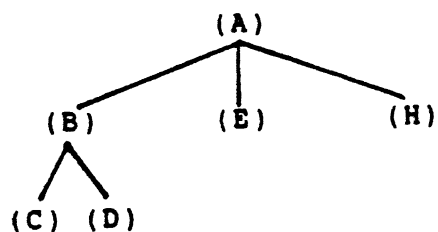


2º



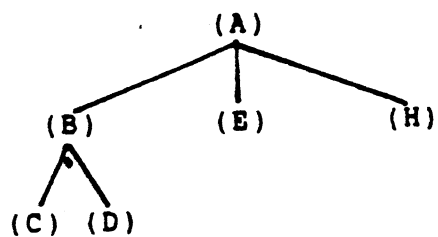
PILA = B E H

3º



PILA = C D E H

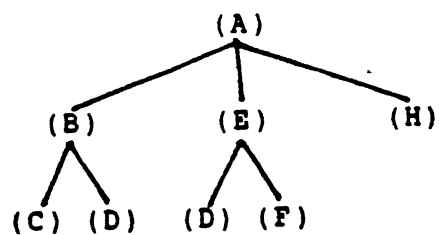
4º



PILA = C D E H

Son callejones
sin salida

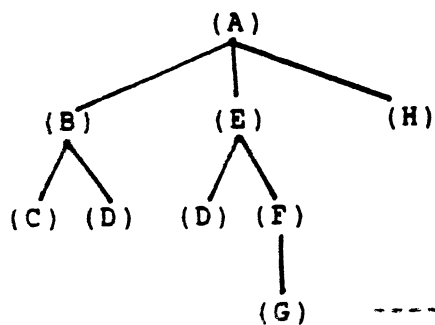
5º



PILA = D F H

No tiene salida

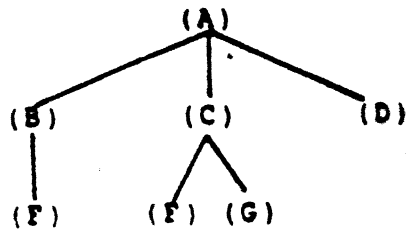
6º



PILA = G H

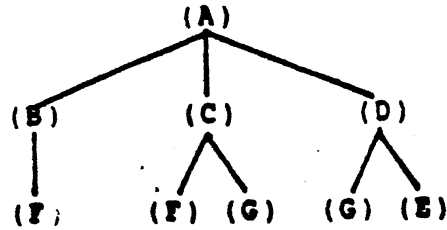
Meta

48



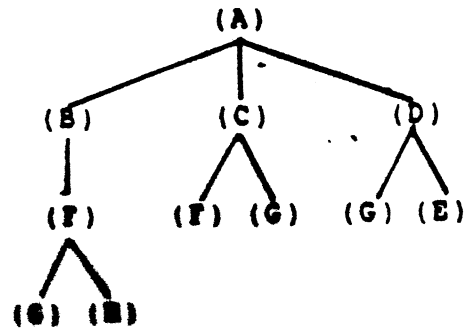
COLA = B F F G

58



COLA = B F G G E

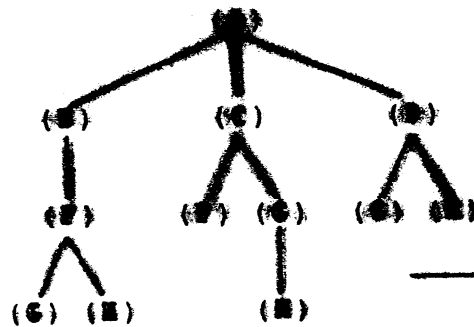
68



COLA = G G G G H

Se cierra porque ya ha
sido expandido.

78



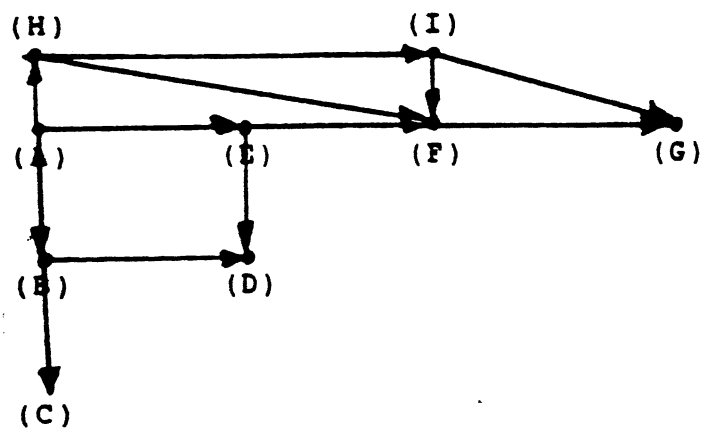
COLA = G G G H H

_____ Callejón sin salida
 _____ Ya se ha expandido

ALGORITMO DE BUSQUEDA EN PROFUNDIDAD

- 1.- Crear una PILA de un sólo elemento, el elemento raíz.
- 2.- Hasta que la pila no esté vacía, o el elemento meta se haya alcanzado; determinar si el primer elemento de la pila es el nodo raíz.
 - 2.a.- Si el primer elemento es el nodo meta, entonces no hacer nada.
 - 2.b.- Si el primer elemento no es el nodo meta, entonces retirar el primer elemento de la pila y añadir (si los hay) los hijos de este elemento a la pila (por delante de todos los demás).
- 3.- Si se ha encontrado el nodo meta anunciar éxito y en caso contrario fracaso.

EJEMPLO



SOLUCION

12

(A)

PILA = A

ALGORITMO DE BUSQUEDA A LO ANCHO

1.- Construir una cola de un elemento, elemento raíz.

2.- Hasta que la cola esté vacía o se haya llegado a la meta, determinar si el primer elemento de la cola es el nodo meta.

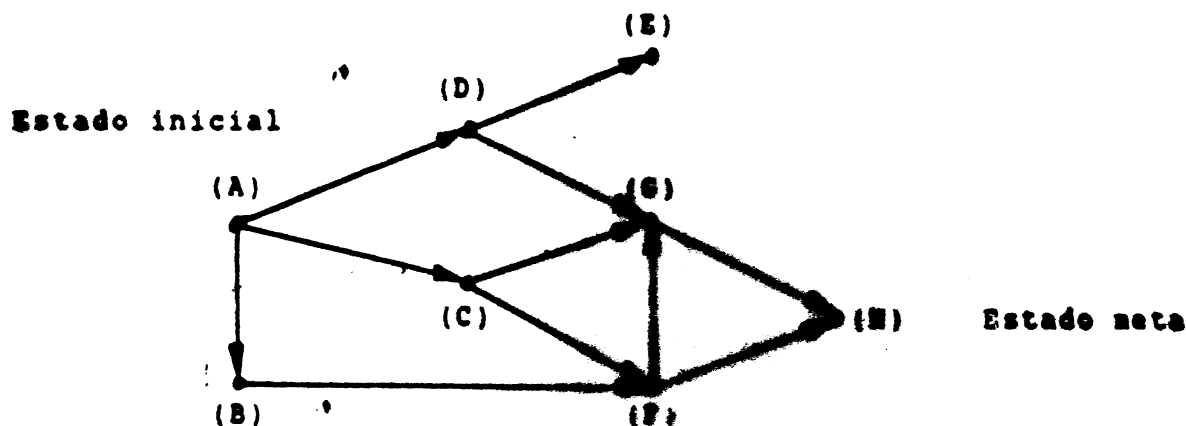
2.a.- Si el elemento es el nodo meta, entonces se hace nada.

2.b.- Si el elemento no es el nodo meta, entonces retirar el primer elemento de la cola y añadir los hijos de este primer elemento (si los tiene) a la cola (por detrás de los demás elementos).

3.- Si se ha encontrado el nodo meta anunciar éxito y en caso contrario fracaso.

EJEMPLO

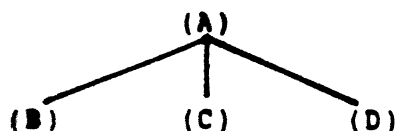
Supongamos que tenemos el siguiente mapa de carreteras y que estamos situados en el punto A y que queremos llegar a H.



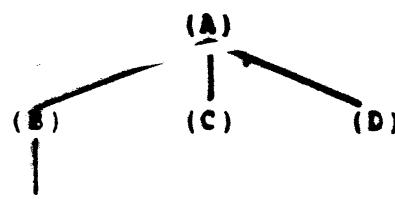
SOLUCION

1º (A) COLA = A

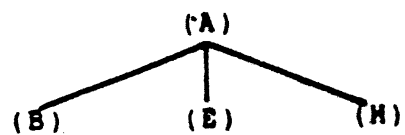
2º (A) COLA = B C D



3º (A) COLA = C D F

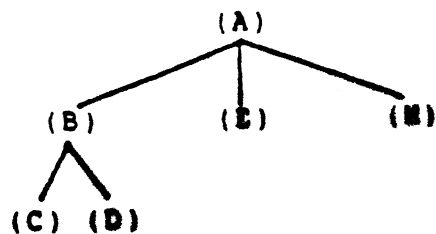


20



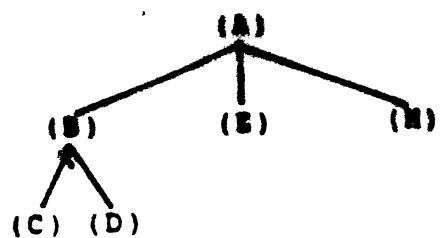
PILA = B E H

30



PILA = C D E H

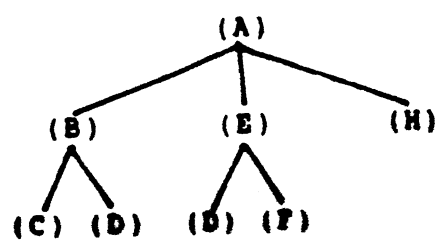
40



PILA = C D E H

Son callejones
sin salida

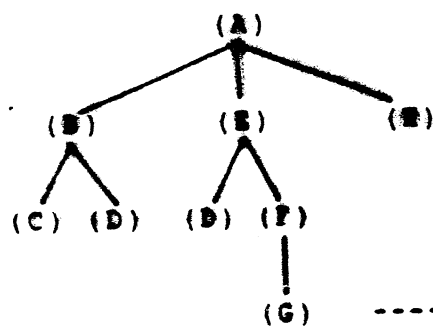
50



PILA = D F H

No tiene salida

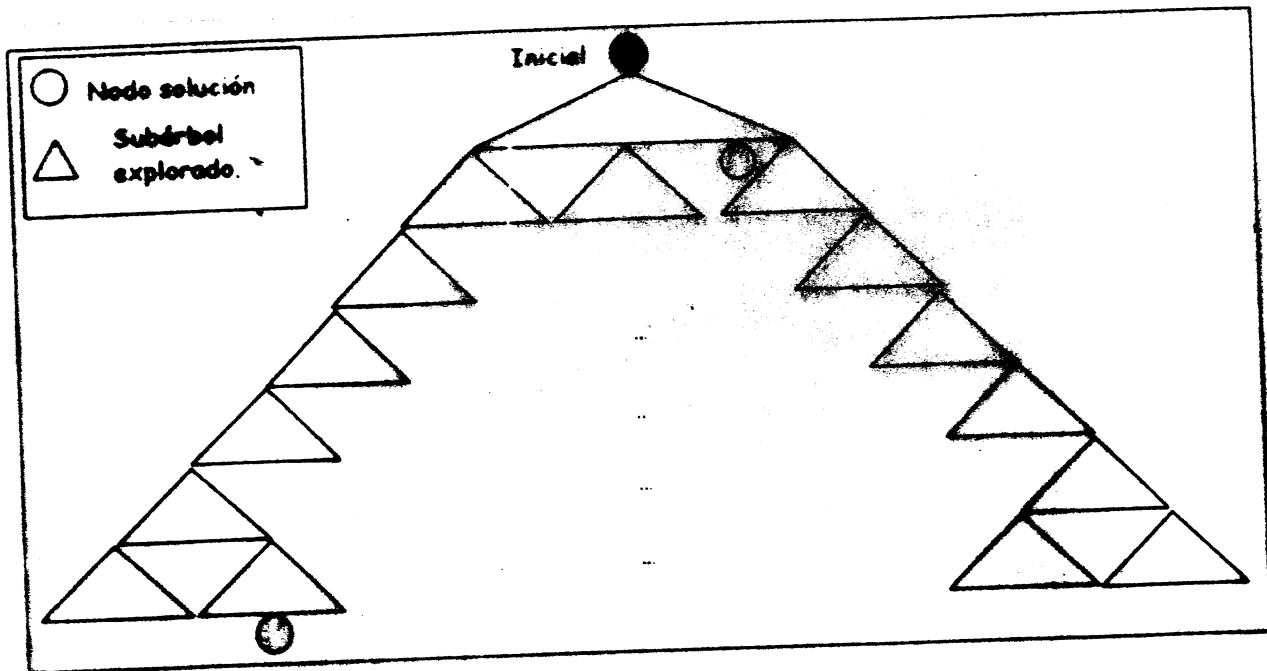
60



PILA = G H

----- Meta

algoritmo
 → ancho
 → profundidad
 → selección



- Hay, como se ve:
 - Una solución profunda.
 - Una solución superficial.
- Las soluciones se evaluarán, según el esfuerzo que representan (esfuerzo que representa de "resolución", no de búsqueda) de la siguiente manera:
 - La solución poco profunda es óptima (en el sentido de mínimo número de procesos para encontrarla, podemos suponer el mismo coste para cada proceso) y representa, por tanto, poco esfuerzo para resolver el problema.
 - La solución más profunda es más costosa porque implica la aplicación de más procesos.

2.3.1.b.b.d.2 Esquema de un árbol normal con dos soluciones y comparación del esfuerzo para encontrarlas.

- A continuación, sobre el mismo esquema de árbol de antes, compararemos lo que cuesta encontrar cada una de las soluciones por cada uno de los métodos vistos.

2.3.1.b.b.d.2.a Búsqueda ciega en amplitud.

- Solución poco profunda:
 - El esfuerzo en encontrarla, por el diseño del propio método, es poco. Como los niveles se recorren enteros no se corre el riesgo de profundizar por ramas que no lleven a la solución.
- Solución profunda:
 - Dado que se recorren los niveles enteros antes de pasar a los siguientes, para encontrar la solución profunda hay que recorrer absolutamente todos los nodos anteriores (de menor profundidad) y, por tanto, el esfuerzo es muy alto.

2.3.1.b.b.d.2.b Búsqueda ciega en profundidad.

- Solución poco profunda.
 - Excepto si, por fortuna, se profundiza exactamente por la rama que lleva a la solución, es probable que se tenga que recorrer alguna otra rama entera antes de