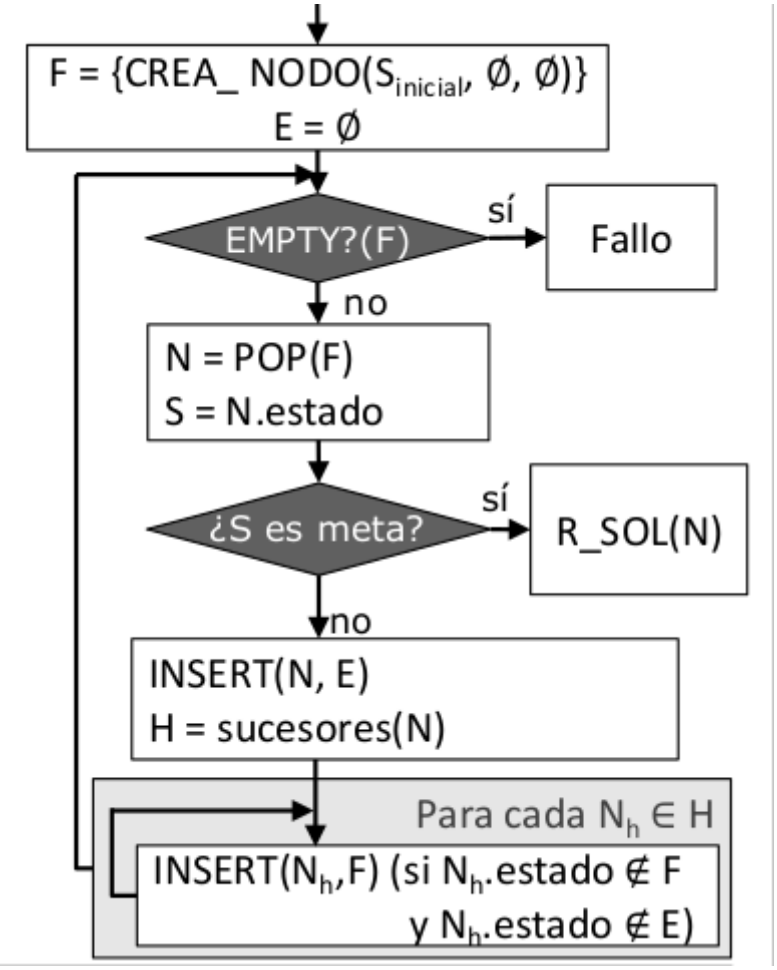
Tema 2-Búsqueda

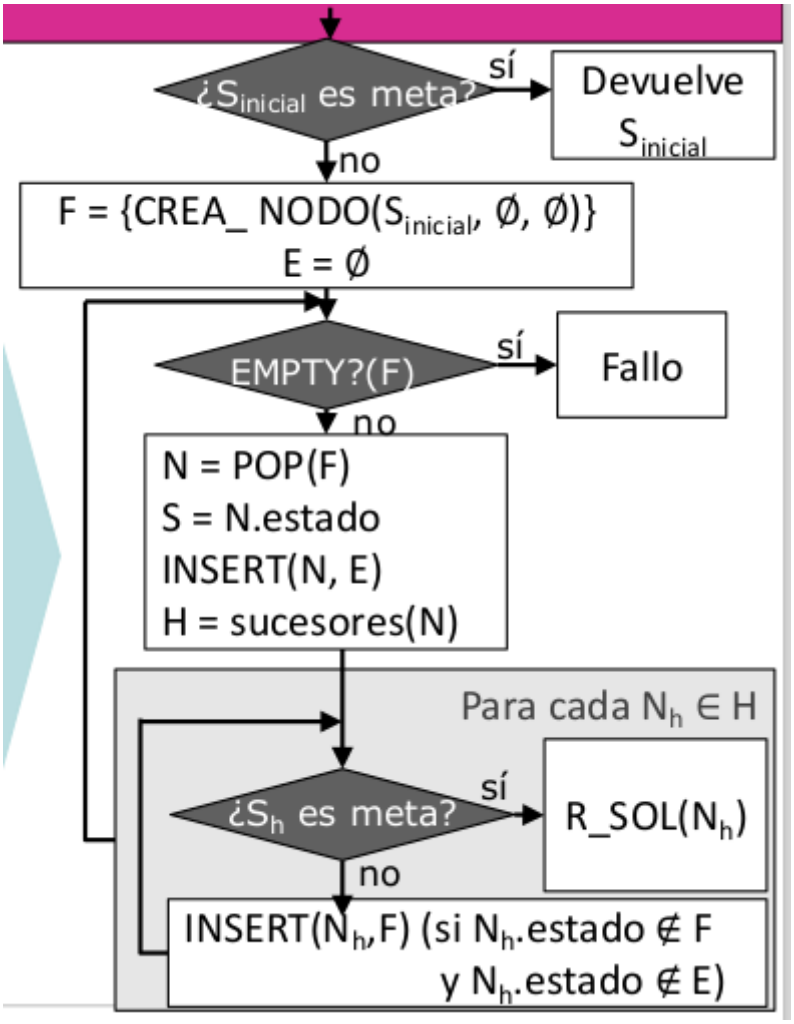
Búsqueda en amplitud

Se usa una cola para la frontera. Se expande el nodo que entró hace más tiempo, es decir, el nodo menos profundo.



El efecto que deja este algoritmo es que explora un nivel de profundidad al completo antes de pasar al siguiente. Siempre encuentra la solución que requiere menos pasos si tiene memoria y tiempo suficientes.

Se podría mejorar ahorrando recursos de esta manera:

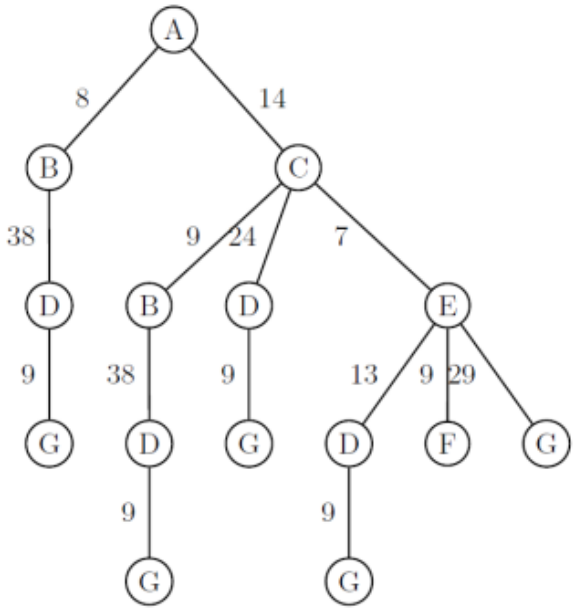


La complejidad temporal de esta es de $O(b^d)$, siendo b el factor de ramificación y d la profundidad de la meta. La complejidad espacial es también de $O(b^d)$.

Es una búsqueda completa, pero solo es óptima si todos los pasos cuestan lo mismo. En general no, porque el nodo más cercano no tiene por qué ser el óptimo.

Un ejemplo de búsqueda en anchura:

Paso	Frontera	Explorados
1	A	-
2	$B^{A,8}$, $C^{A,14}$	A
3	$C^{A,14}$, $D^{B,46}$	A, $B^{A,8}$
4	$D^{B,46}$, $E^{C,21}$	A, $B^{A,8}$, $C^{A,14}$
5	$E^{C,21}$, $\mathbf{G^{D,55}}$	A, $B^{A,8}$, $C^{A,14}$, $D^{B,46}$



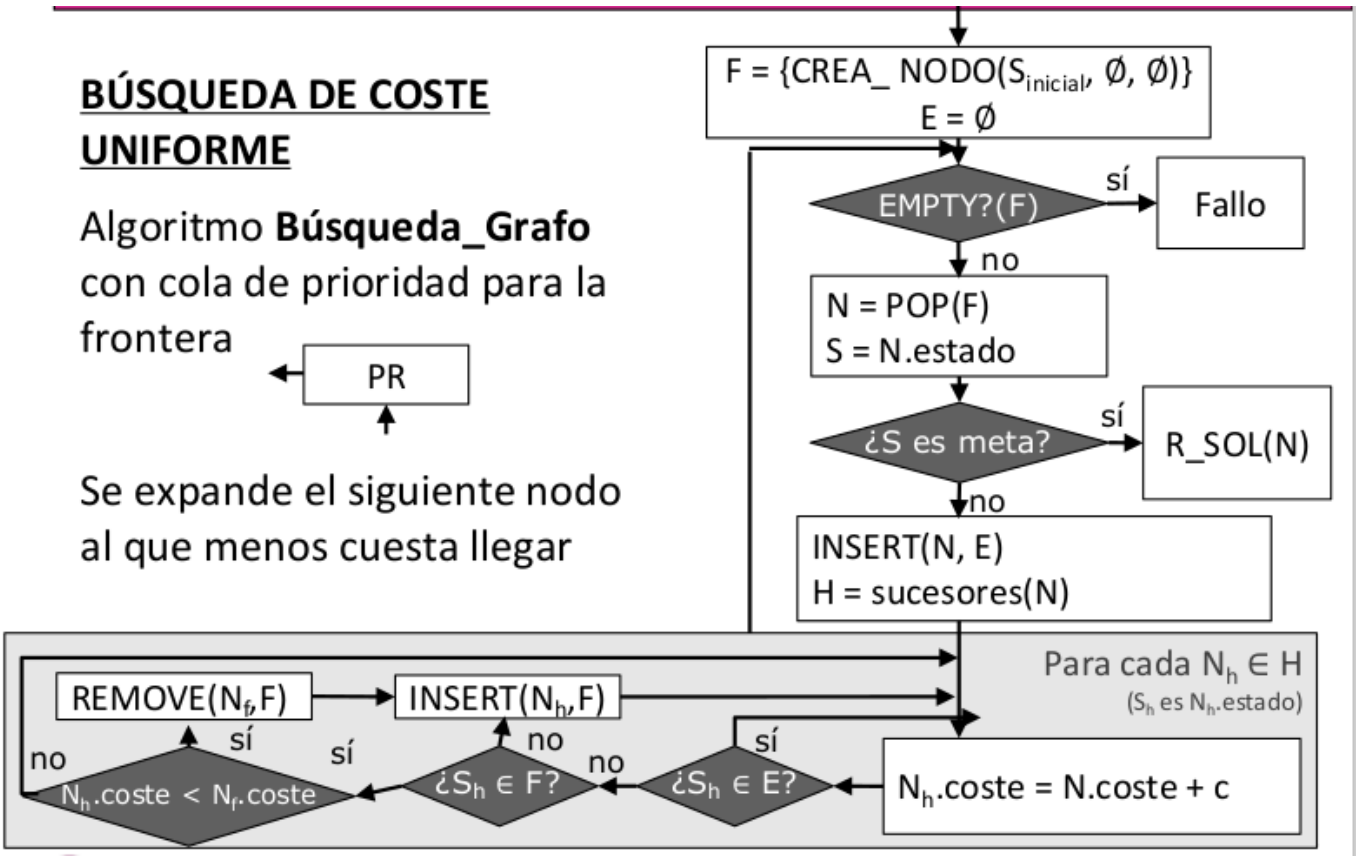
Notación nodos:

$S^{p,c}$ donde

S es el estado
p es el estado padre
c es el coste del camino A->S

Búsqueda de coste uniforme

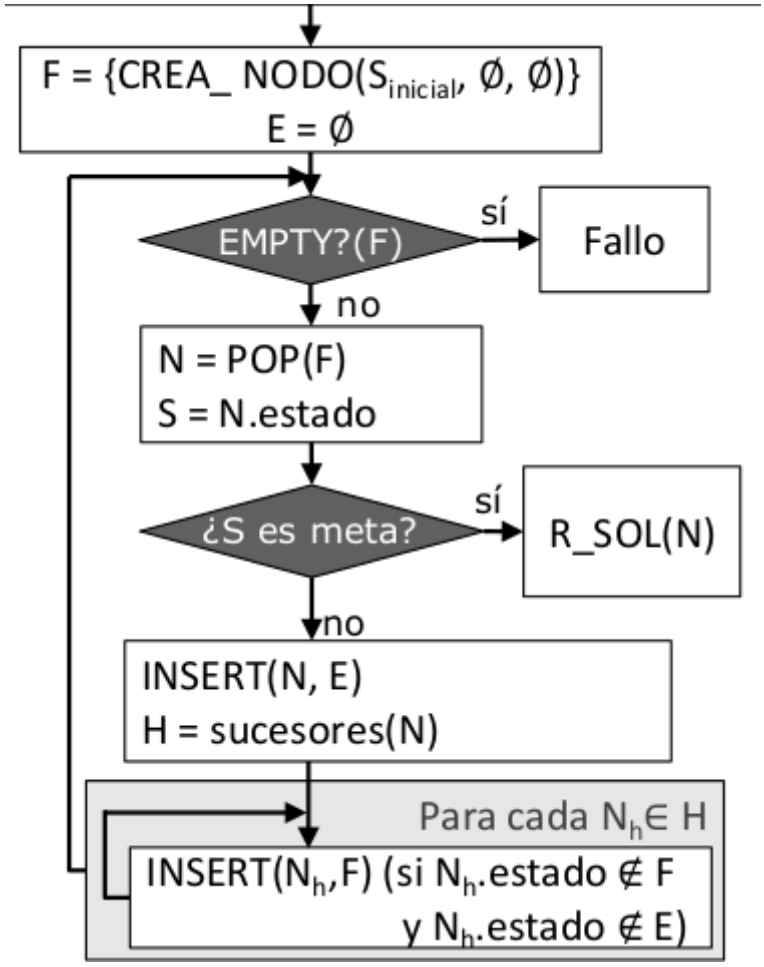
La cola de la frontera pasa a ser una cola de prioridad ordenada por coste. Se expande siempre el camino con menor coste.



Siempre obtiene un solución óptima.

Búsqueda en profundidad

Se usa una pila para la frontera. Se expande el nodo que entró hace menos tiempo, es decir, se sigue cada camino hasta el final antes de cambiar.



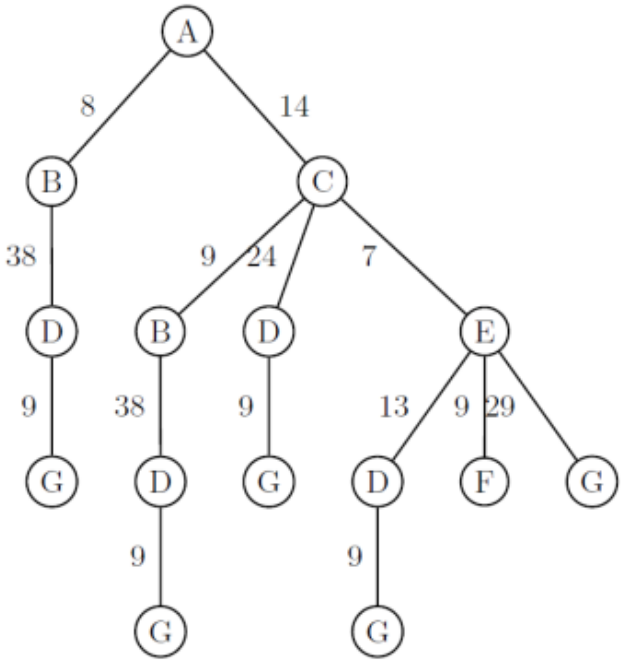
Es completa, pero no es óptima, ya que puede encontrar una solución más profunda que otra que esté en una rama no expandida.

La complejidad temporal es de $O(b^m)$, siendo b la ramificación y m la profundidad máxima. La complejidad espacial es de $O(b * m)$.

Un ejemplo de búsqueda en profundidad:

LIFO

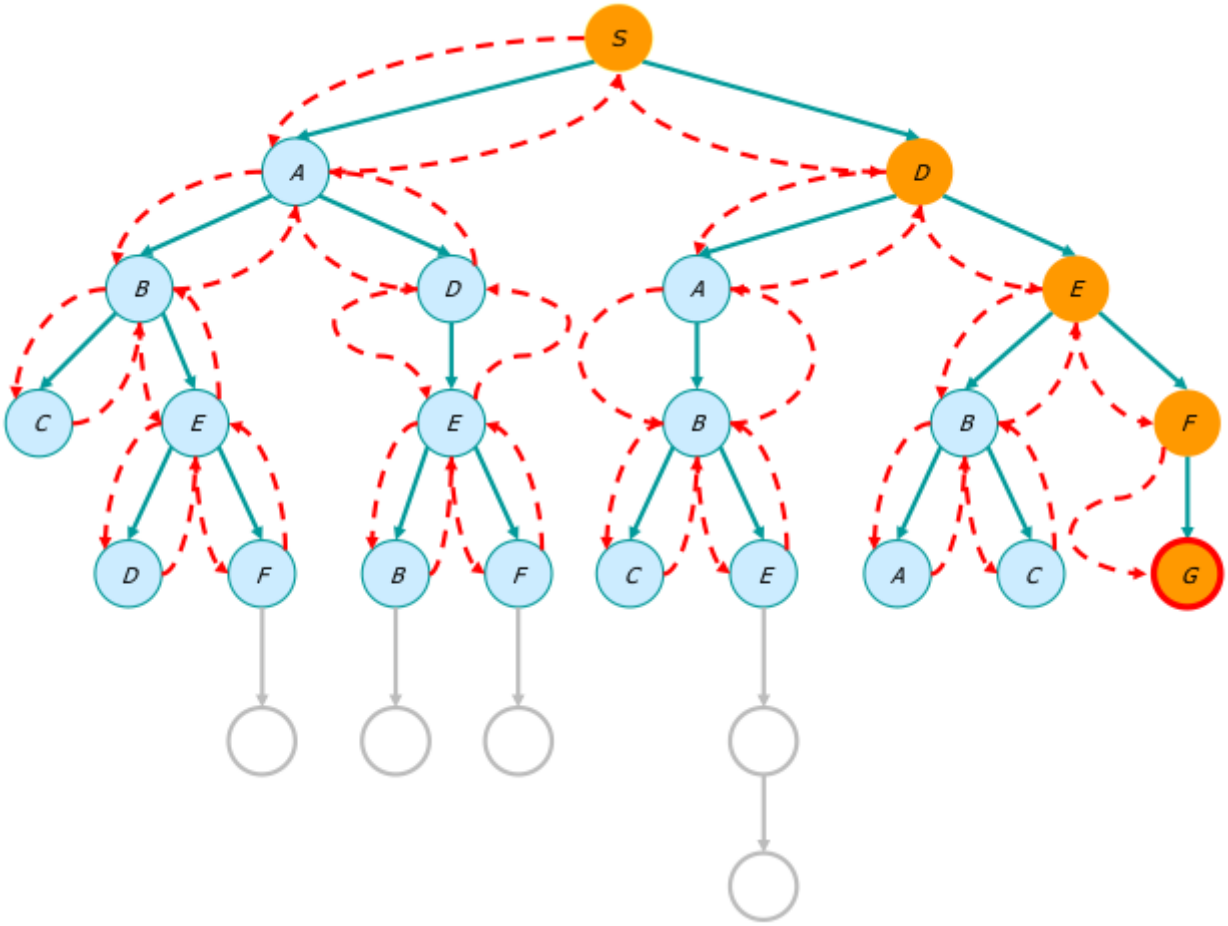
#	Frontera	Explorados
1	A	-
2	$B^{A,8}$, $C^{A,14}$	A
3	$D^{B,46}$, $C^{A,14}$	A, $B^{A,8}$
4	$\mathbf{G^{D,55}}$, $C^{A,14}$	A, $B^{A,8}$, $D^{B,46}$



Se puede mejorar con backtracking. Consiste en que una vez se explora un nodo hasta el final y se descarta, se elimina para ahorrar memoria. Esto mejora la complejidad espacial, que pasa a ser $O(m)$.

Búsqueda de profundidad limitada

Consiste en poner un límite a la profundidad máxima que se puede explorar para evitar caer en caminos infinitos. Por ejemplo:



Esta engloba las ventajas de la búsqueda en profundidad y de la búsqueda en anchura.