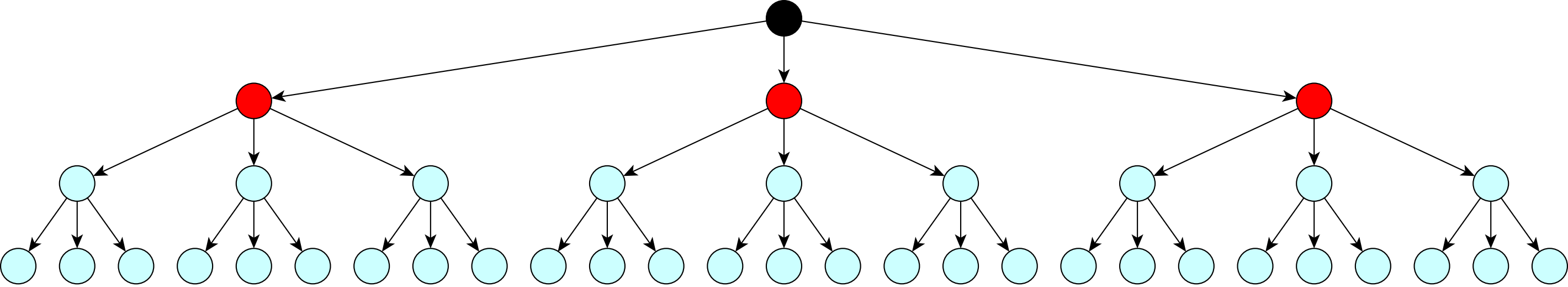


Algoritmo DFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0

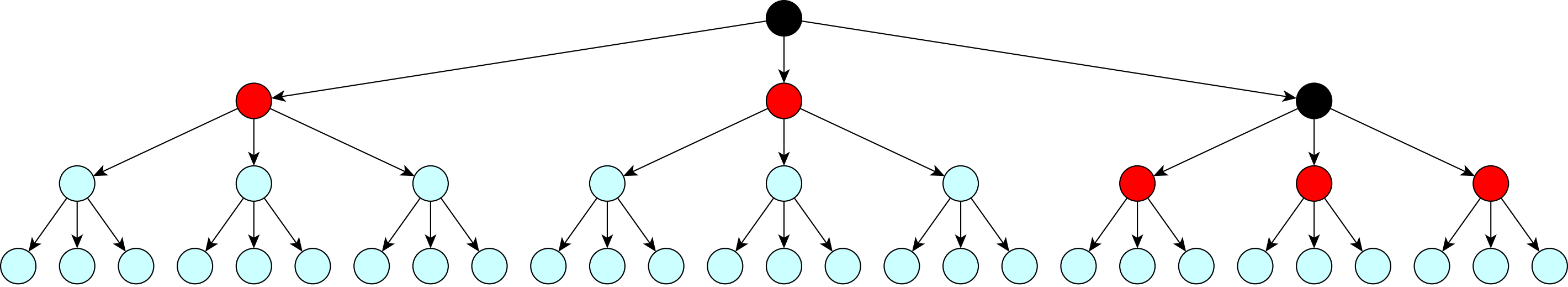


Algoritmo DFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0
1	3	1

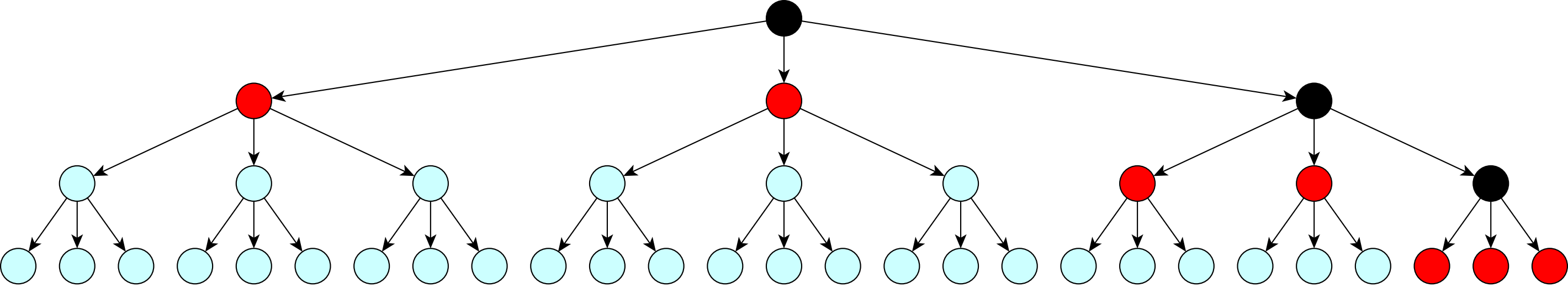


Algoritmo DFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0
1	3	1
2	5	2



Algoritmo DFS

1. Memoria.

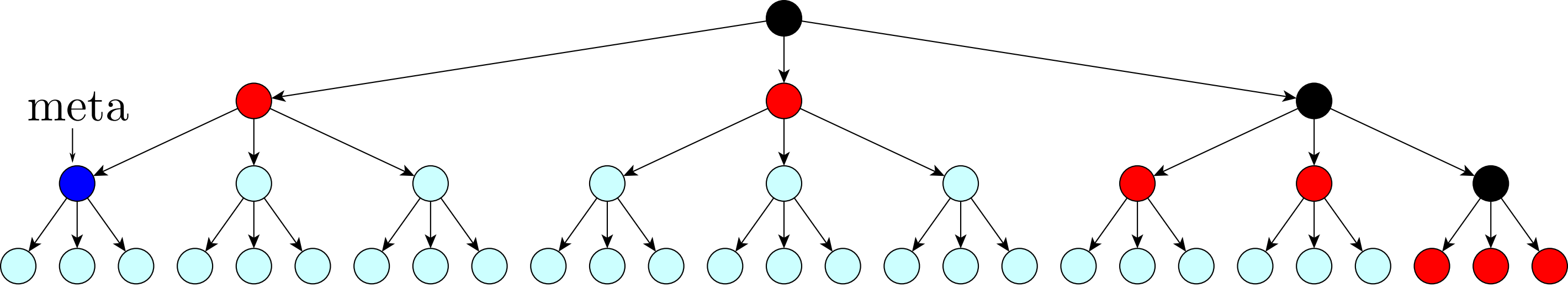
Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0
1	3	1
2	5	2
3	7	3



- $O(bn)$

2. Tiempo.



Algoritmo DFS

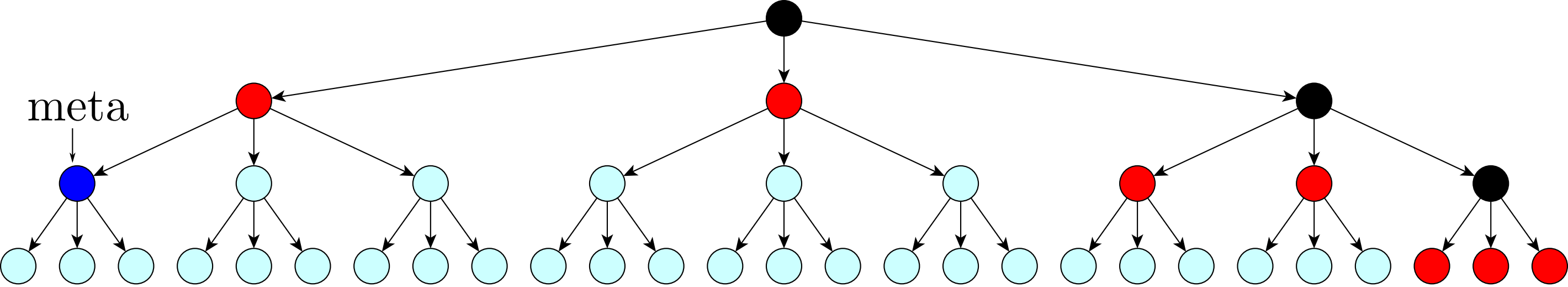
1. Memoria.

- $O(bn)$

2. Tiempo.

Peor escenario:

La meta esta en última rama
del árbol de búsqueda,
a una profundidad d



Algoritmo DFS

1. Memoria.

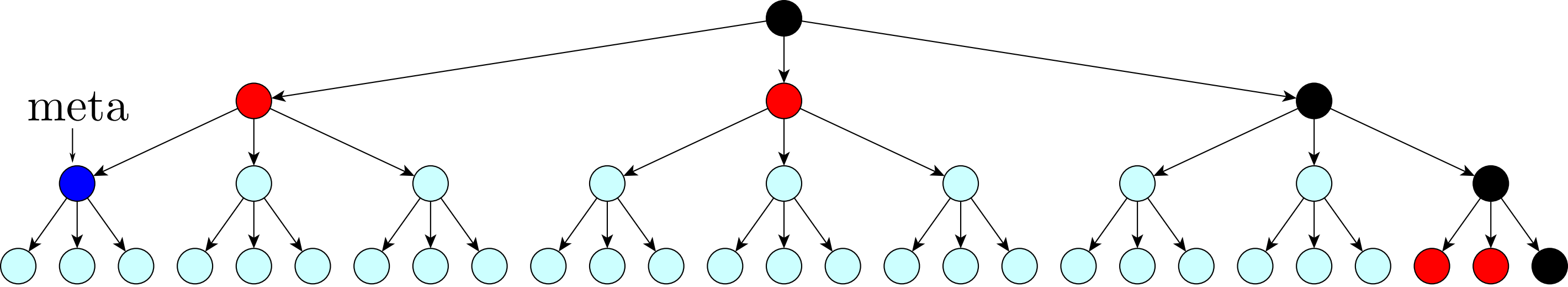
- $O(bn)$

2. Tiempo.

Peor escenario:

La meta esta en última rama
del árbol de búsqueda,
a una profundidad d

Continuamos la búsqueda...



Algoritmo DFS

1. Memoria.

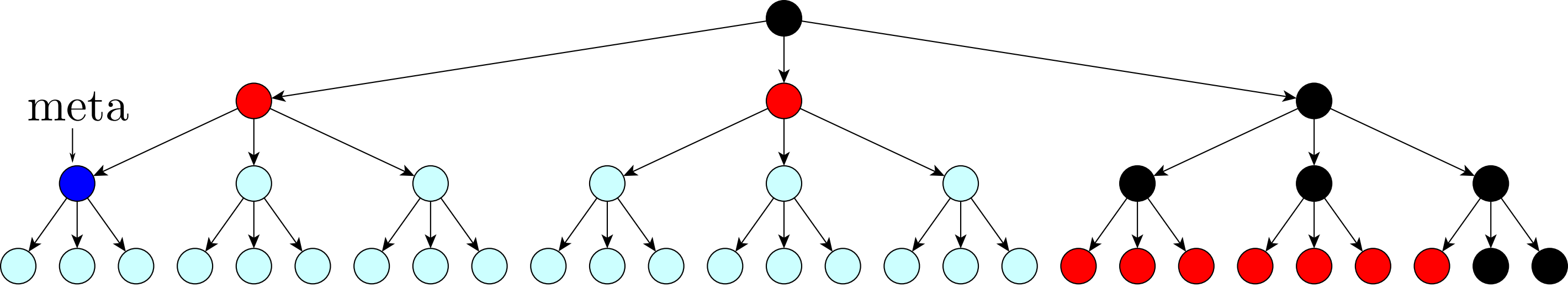
- $O(bn)$

2. Tiempo.

Peor escenario:

La meta esta en última rama
del árbol de búsqueda,
a una profundidad d

el algoritmo alcanza la profundidad m



Algoritmo DFS

1. Memoria.

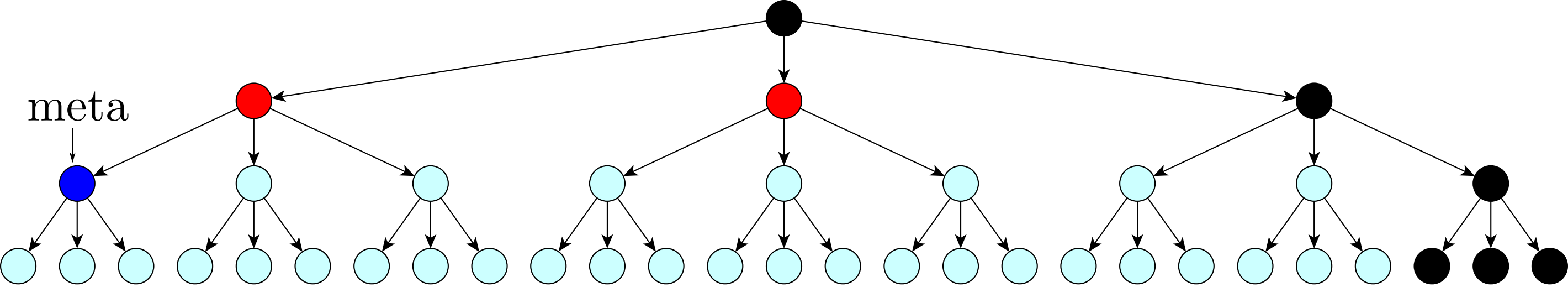
- $O(bn)$

2. Tiempo.

Peor escenario:

La meta esta en última rama
del árbol de búsqueda,
a una profundidad d

continuamos la búsqueda...



Algoritmo DFS

1. Memoria.

■ $O(bn)$

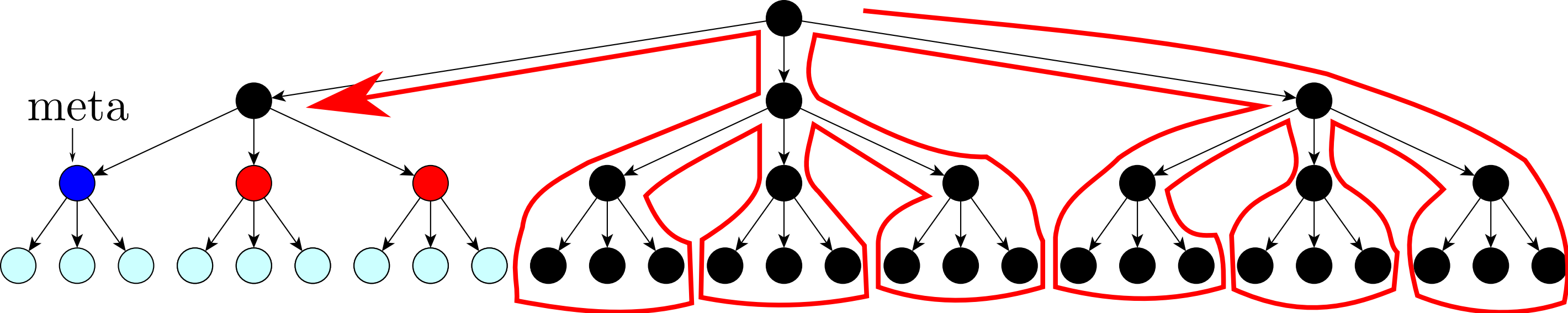
2. Tiempo.

Peor escenario:

La meta está en última rama
del árbol de búsqueda,
a una profundidad d

Tras alcanzar la profundidad máxima
el algoritmo retrocede.

La profundidad de los estados explorados no
crece monótonicamente con las iteraciones.



Algoritmo DFS

1. Memoria.

■ ~~$O(bn)$~~

$O(b^m)$

2. Tiempo.

$O(b^m)$

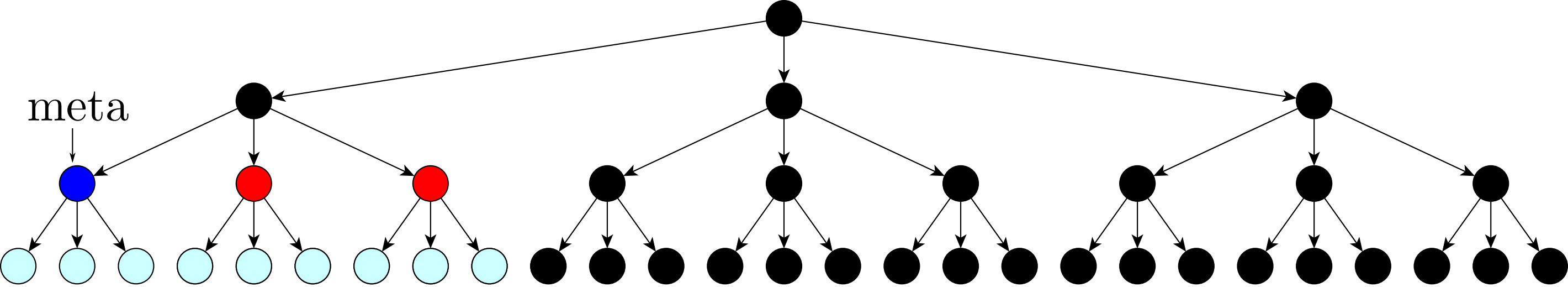
Peor escenario:

La meta esta en última rama
del árbol de búsqueda,
a una profundidad d

prácticamente expandimos todos los niveles
del arbol a profundidad m

cada nivel tiene b^n estados

El conjunto de expandidos contiene todos los estados



Algoritmo DFS

1. Memoria.

- $O(b^m)$

2. Tiempo.

- $O(b^m)$

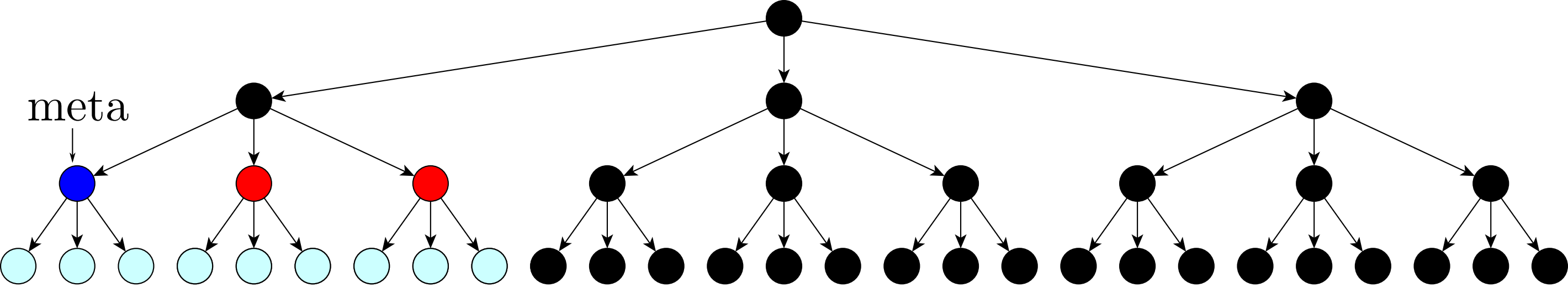
3. Calidad.

- Solución no óptima.

4. Completez.

- No es completo.

Ya habíamos visto que para un grafo DFS no nos garantiza la solución óptima en número de pasos.



Algoritmo DFS

1. Memoria.

- $O(b^m)$

2. Tiempo.

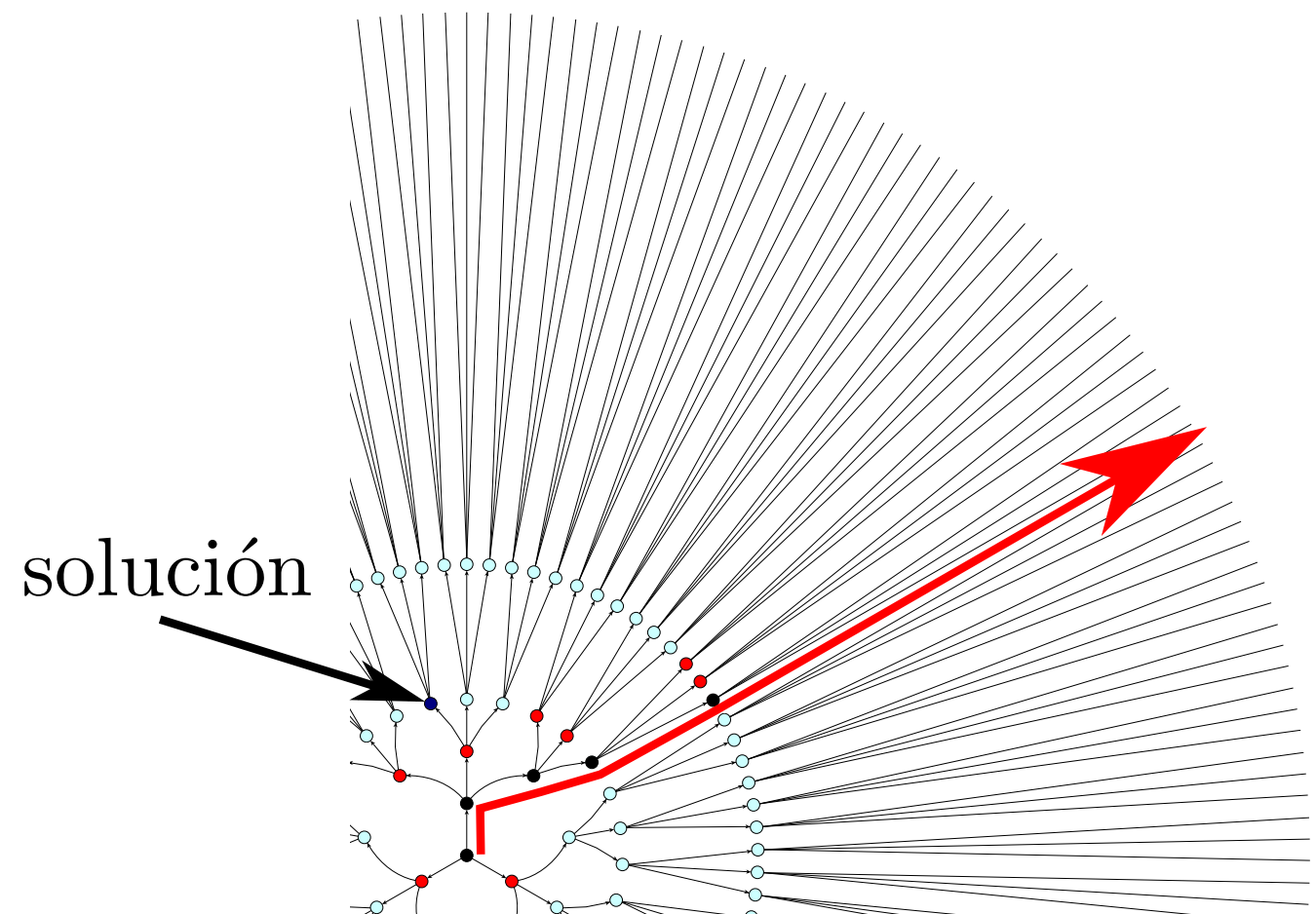
- $O(b^m)$

3. Calidad.

- Solución no óptima.

4. Completez.

- No es completo.



DFS no es completo si el grafo es infinito, a pesar de que la solución pudiera estar relativamente cerca del estado inicial.