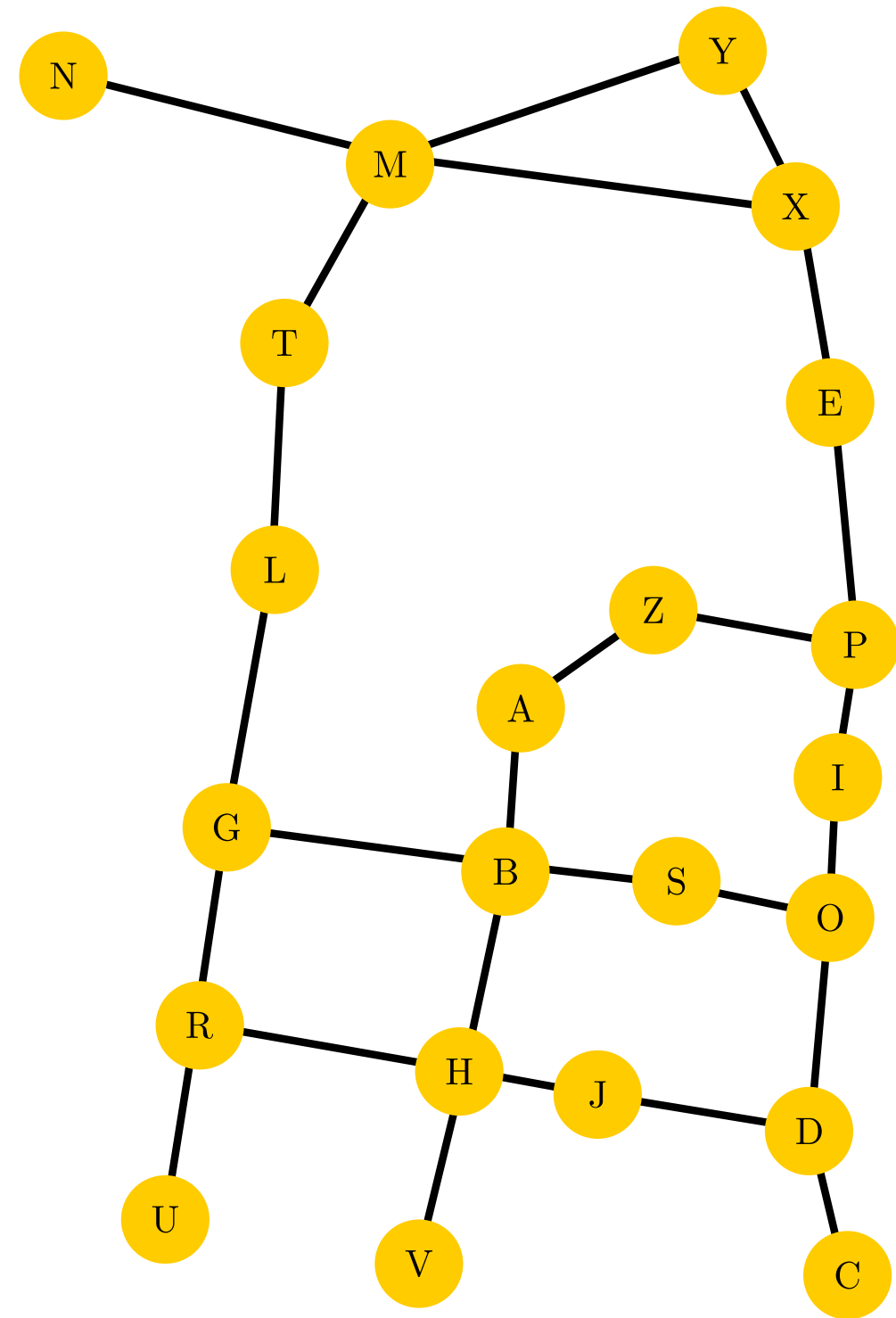
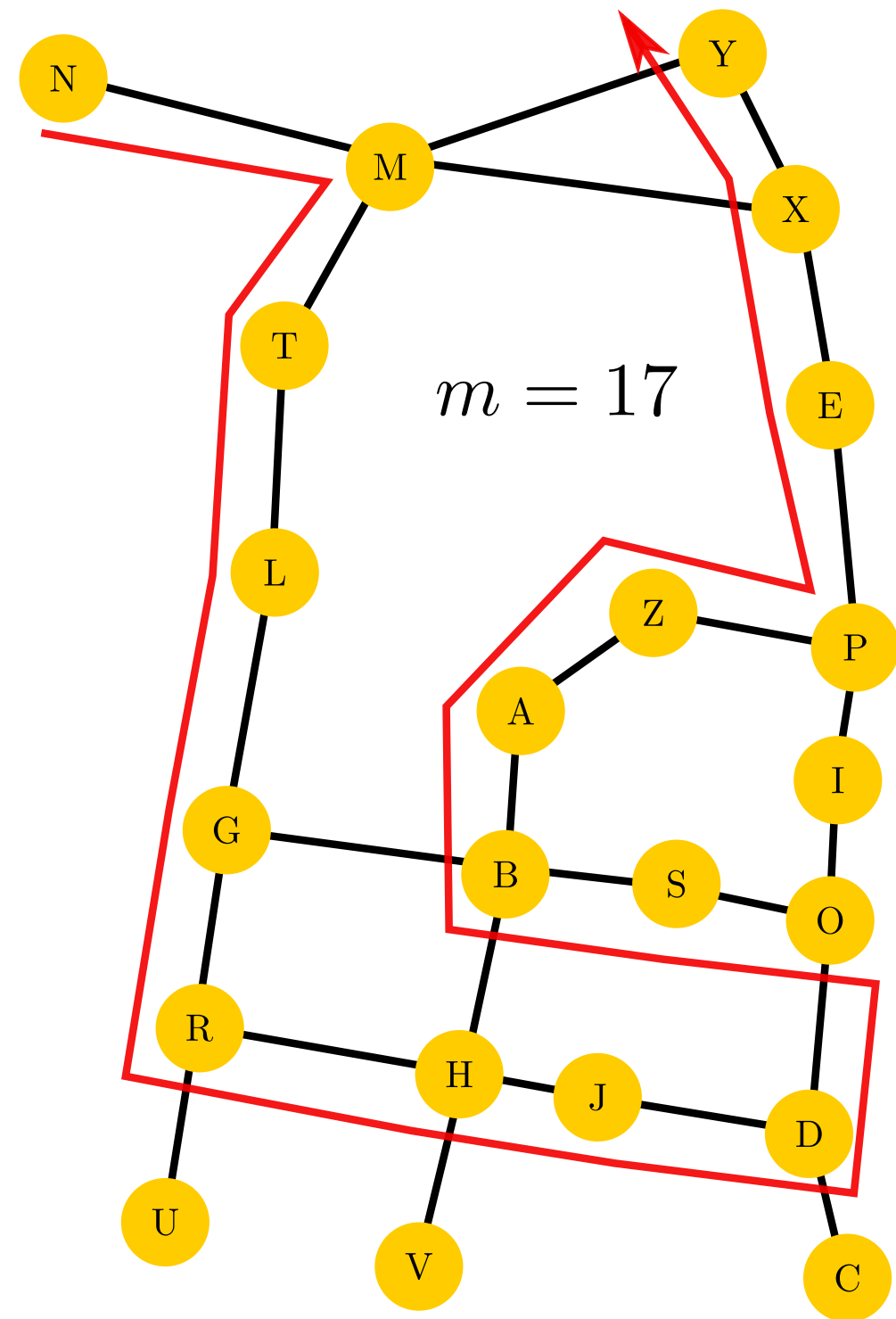


Parametrización del Problema a resolver.



Parametrización del Problema a resolver.

- Longitud de la trayectoria más larga (m)
(sin que se formen ciclos)



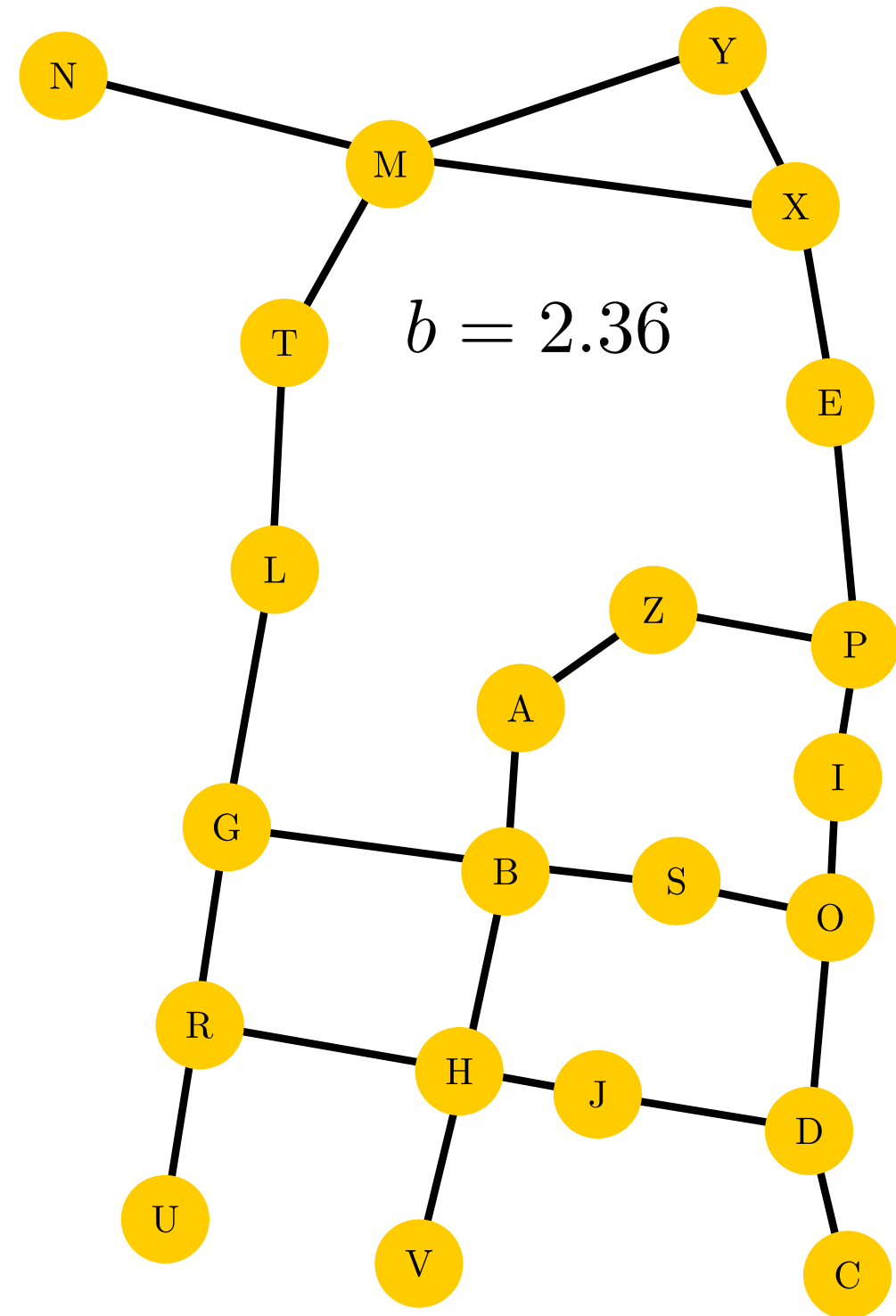
Parametrización del Problema a resolver.

- Longitud de la trayectoria más larga (m)
- Factor de ramificación
(*branching factor*: b).

Número máximo de acciones posibles.

En un grafo no uniforme, el promedio.

$$b = \frac{3 \times 4 + 6 \times 3 + 9 \times 2 + 4 \times 1}{22}$$



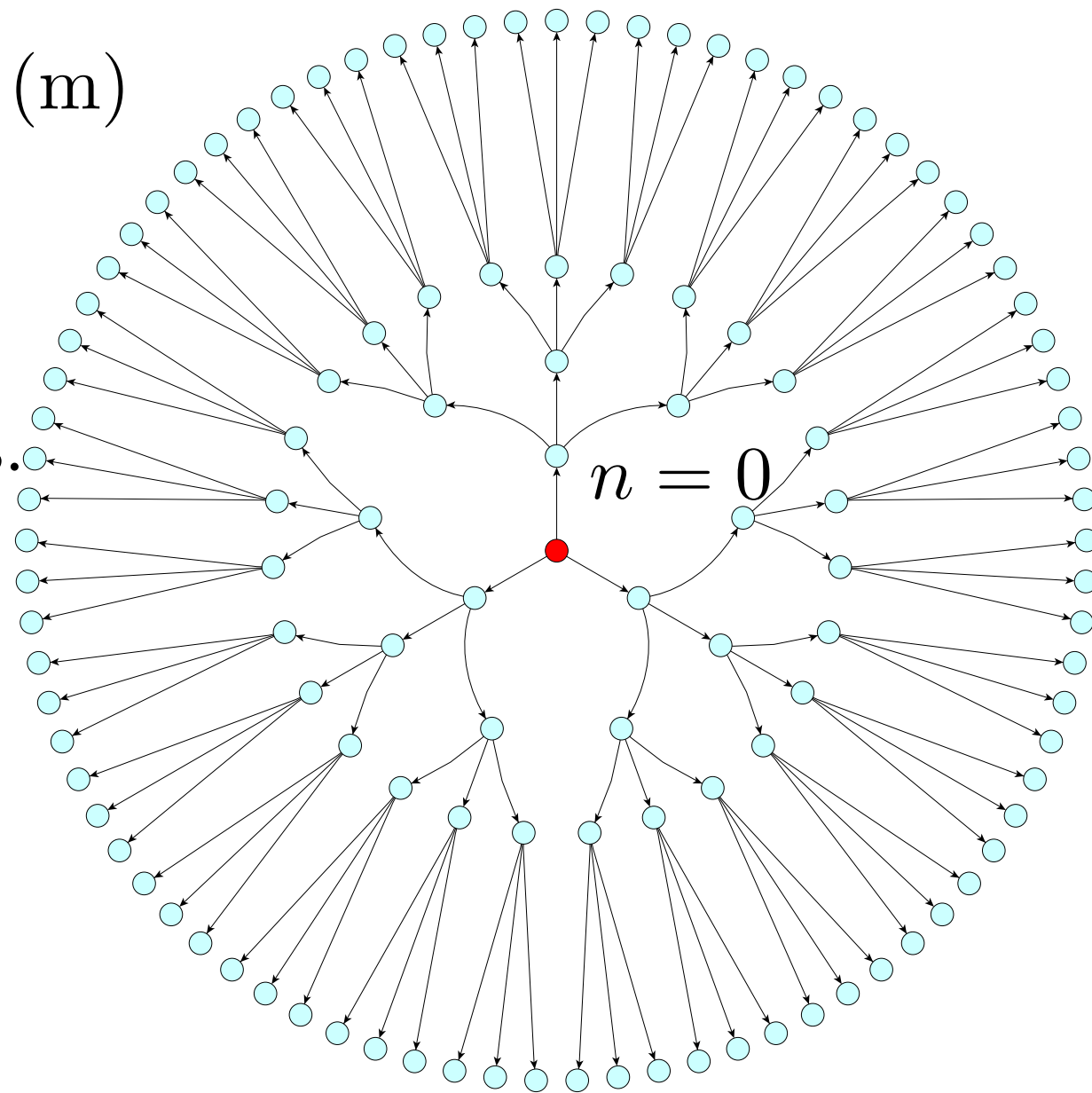
Parametrización del Problema a resolver.

- Longitud de la trayectoria más larga (m)
- Factor de ramificación
(*branching factor*: b).

Número máximo de acciones posibles.

- Profundidad de un vértice (n).

Numero de acciones desde el estado inicial.



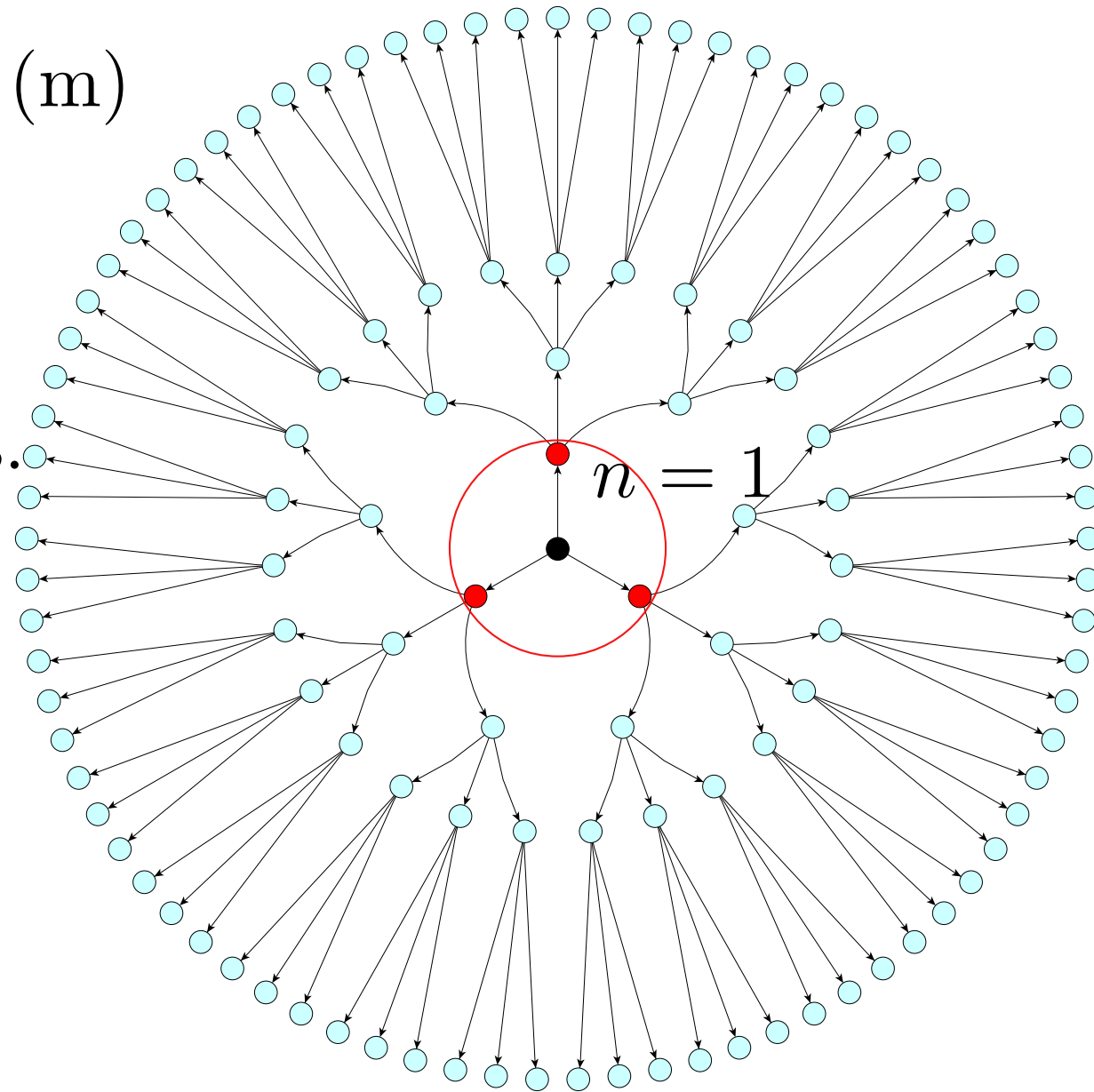
Parametrización del Problema a resolver.

- Longitud de la trayectoria más larga (m)
- Factor de ramificación
(*branching factor: b*).

Número máximo de acciones posibles.

- Profundidad de un vértice (n).

Numero de acciones desde el estado inicial.



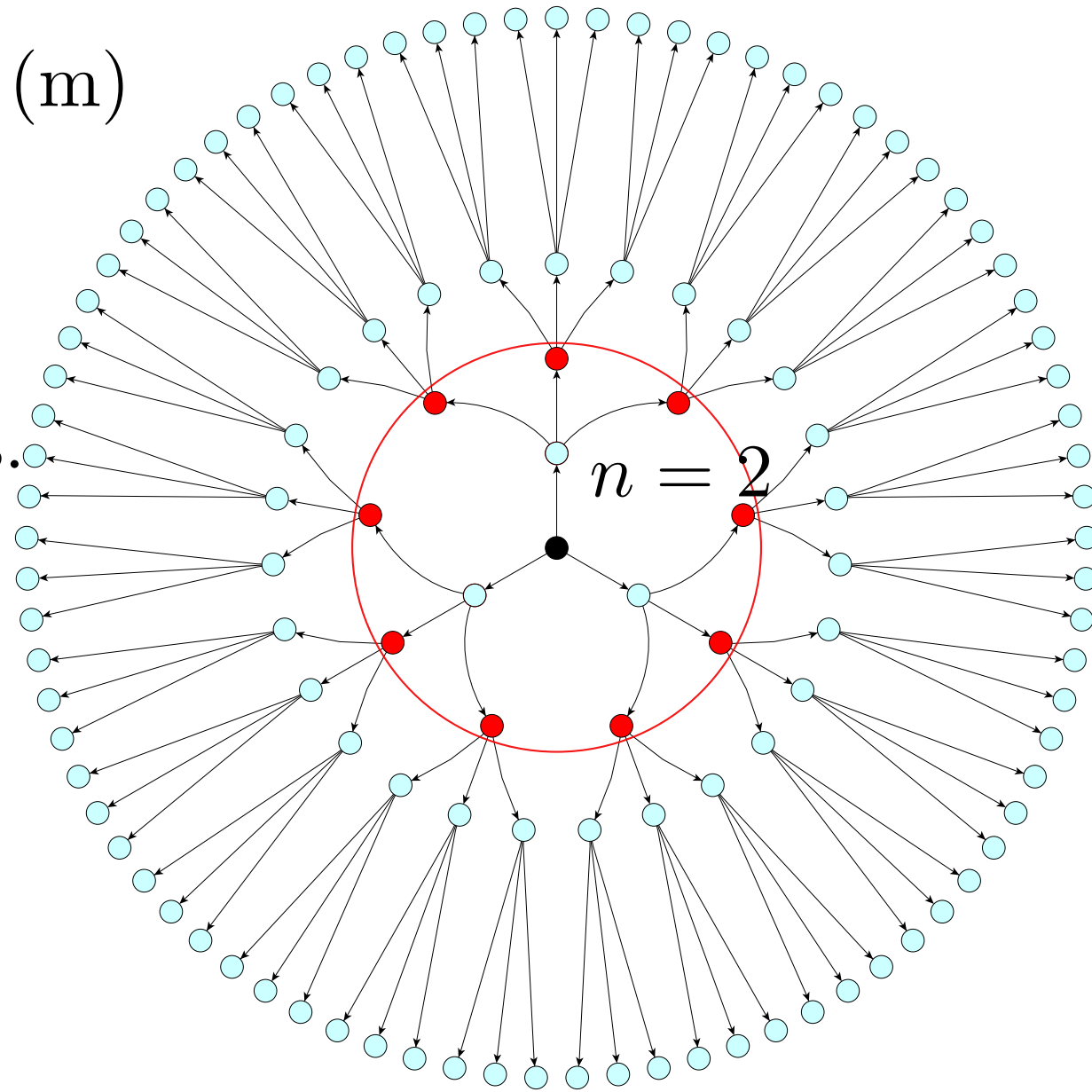
Parametrización del Problema a resolver.

- Longitud de la trayectoria más larga (m)
- Factor de ramificación
(*branching factor: b*).

Número máximo de acciones posibles.

- Profundidad de un vértice (n).

Numero de acciones desde el estado inicial.



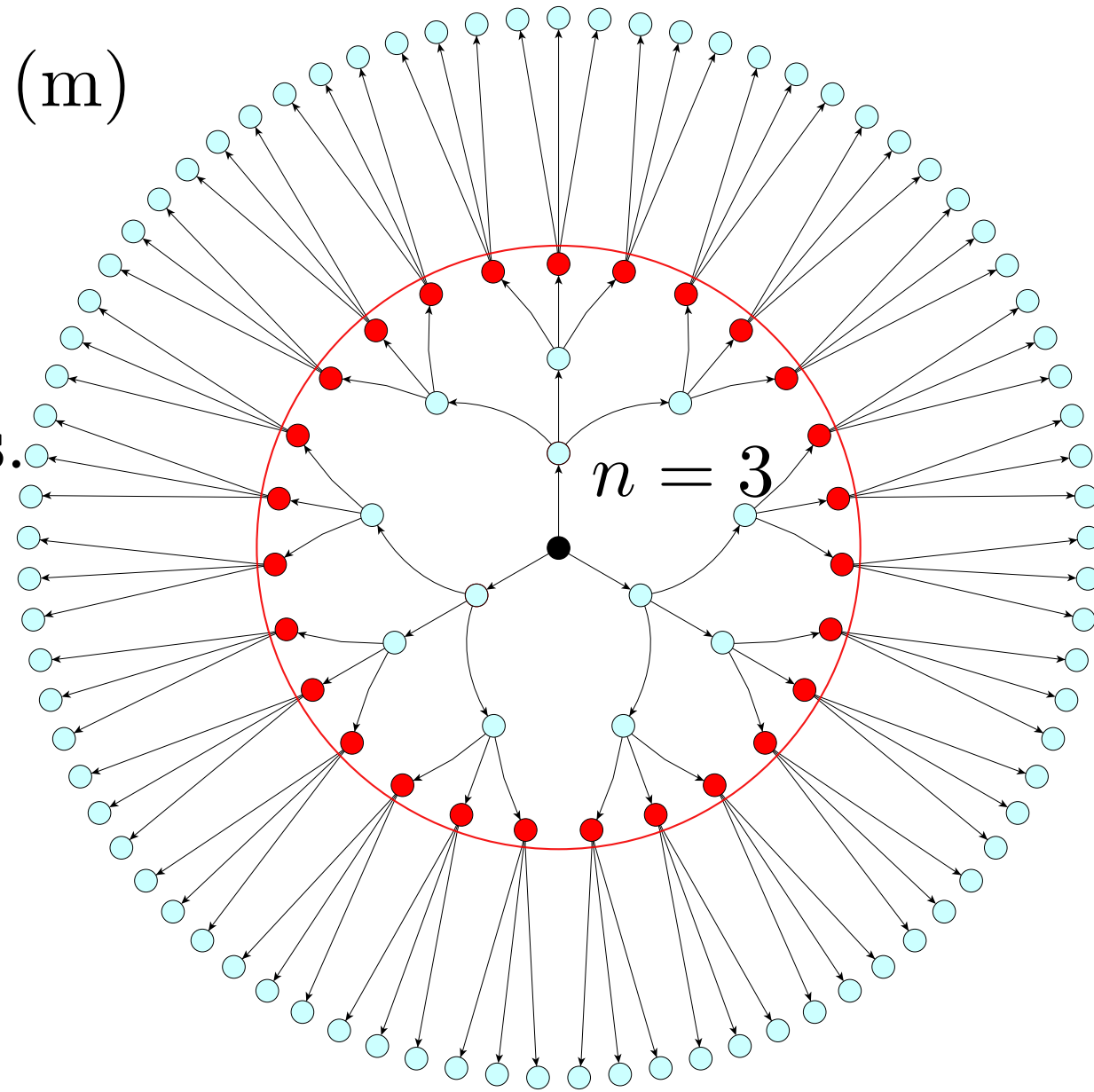
Parametrización del Problema a resolver.

- Longitud de la trayectoria más larga (m)
- Factor de ramificación
(*branching factor: b*).

Número máximo de acciones posibles.

- Profundidad de un vértice (n).

Numero de acciones desde el estado inicial.



Parametrización del Problema a resolver.

- Longitud de la trayectoria más larga (m)

- Factor de ramificación
(*branching factor: b*).

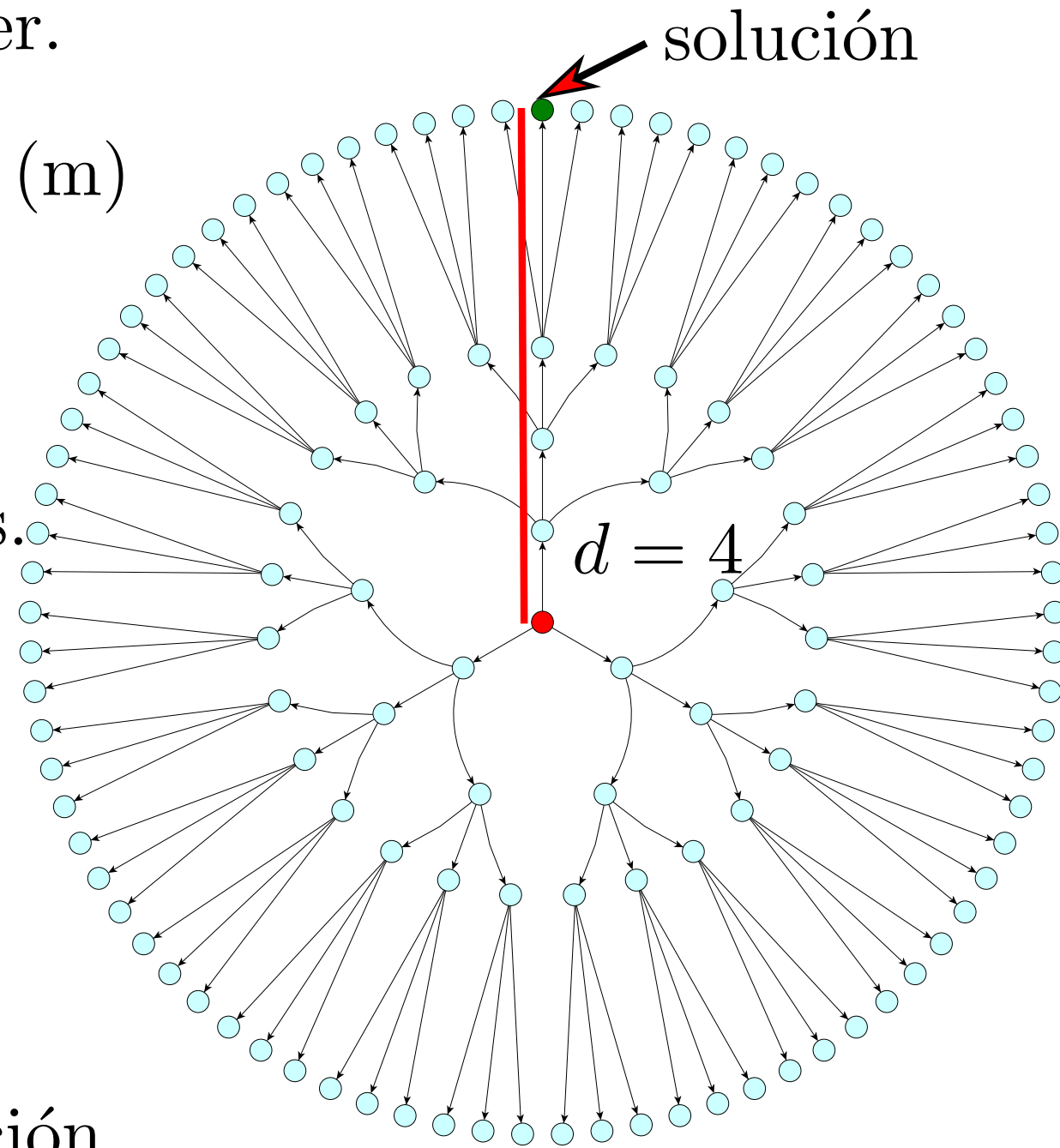
Número máximo de acciones posibles.

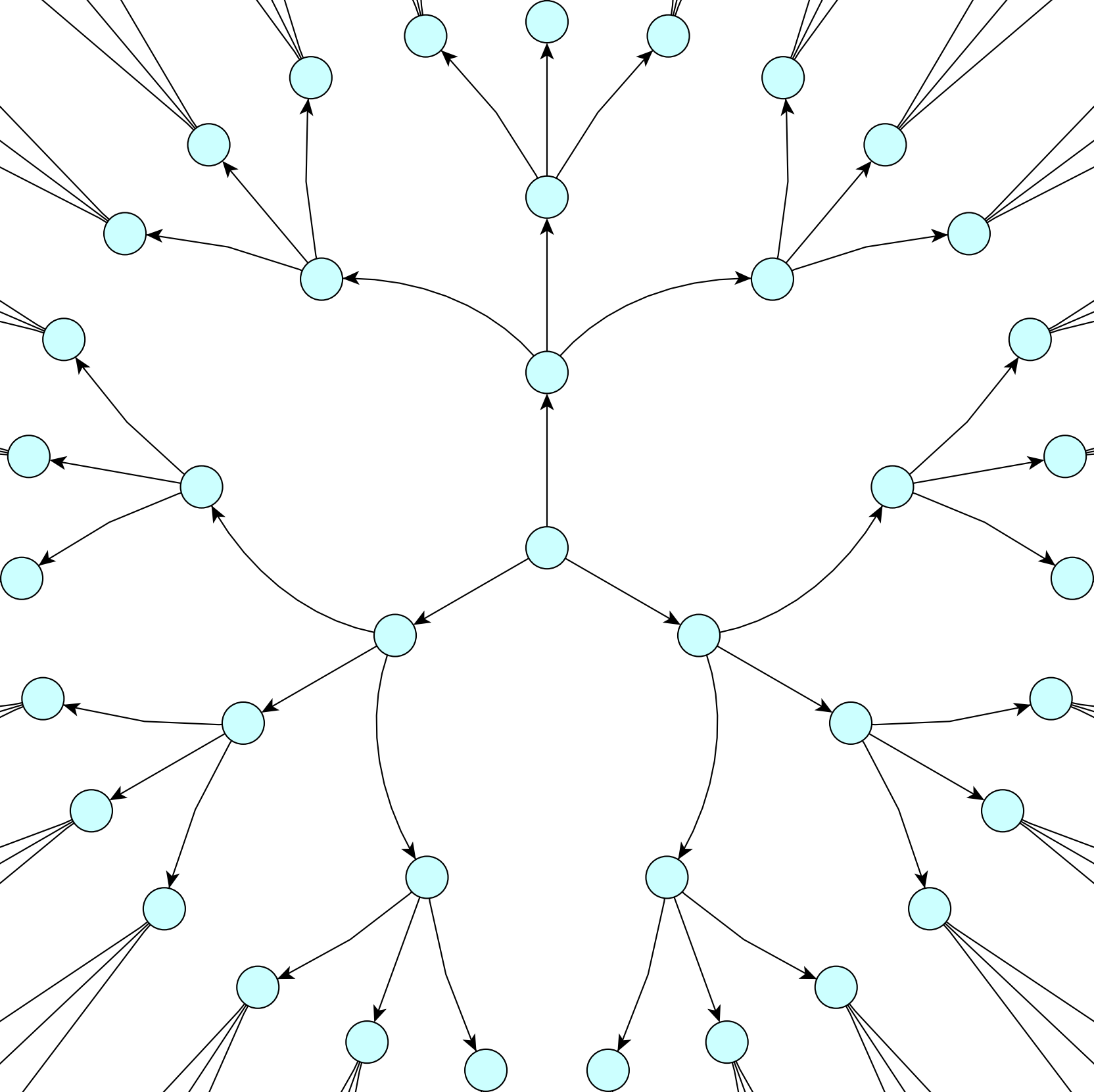
- Profundidad de un vértice (n).

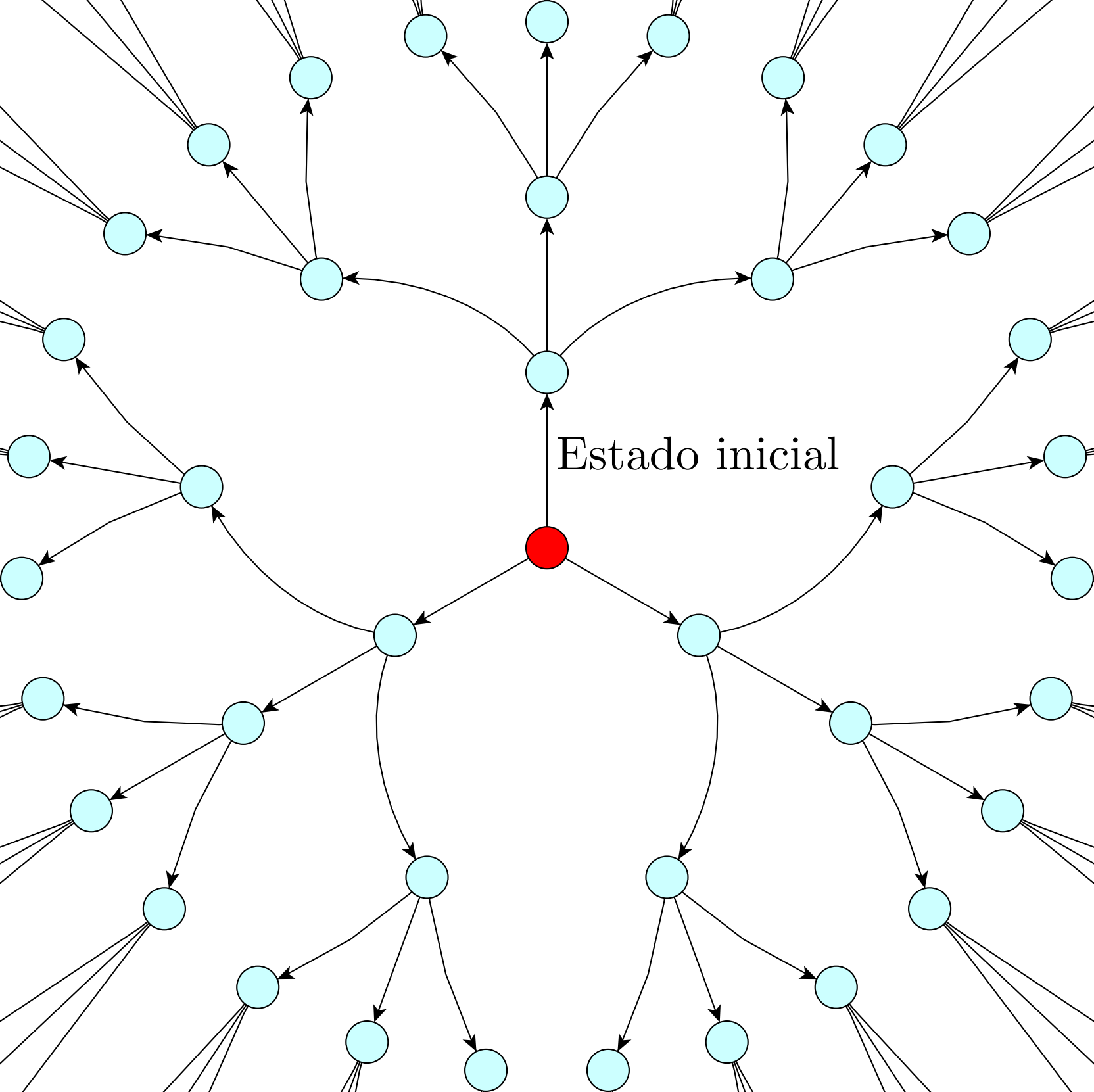
Numero de acciones desde el estado inicial.

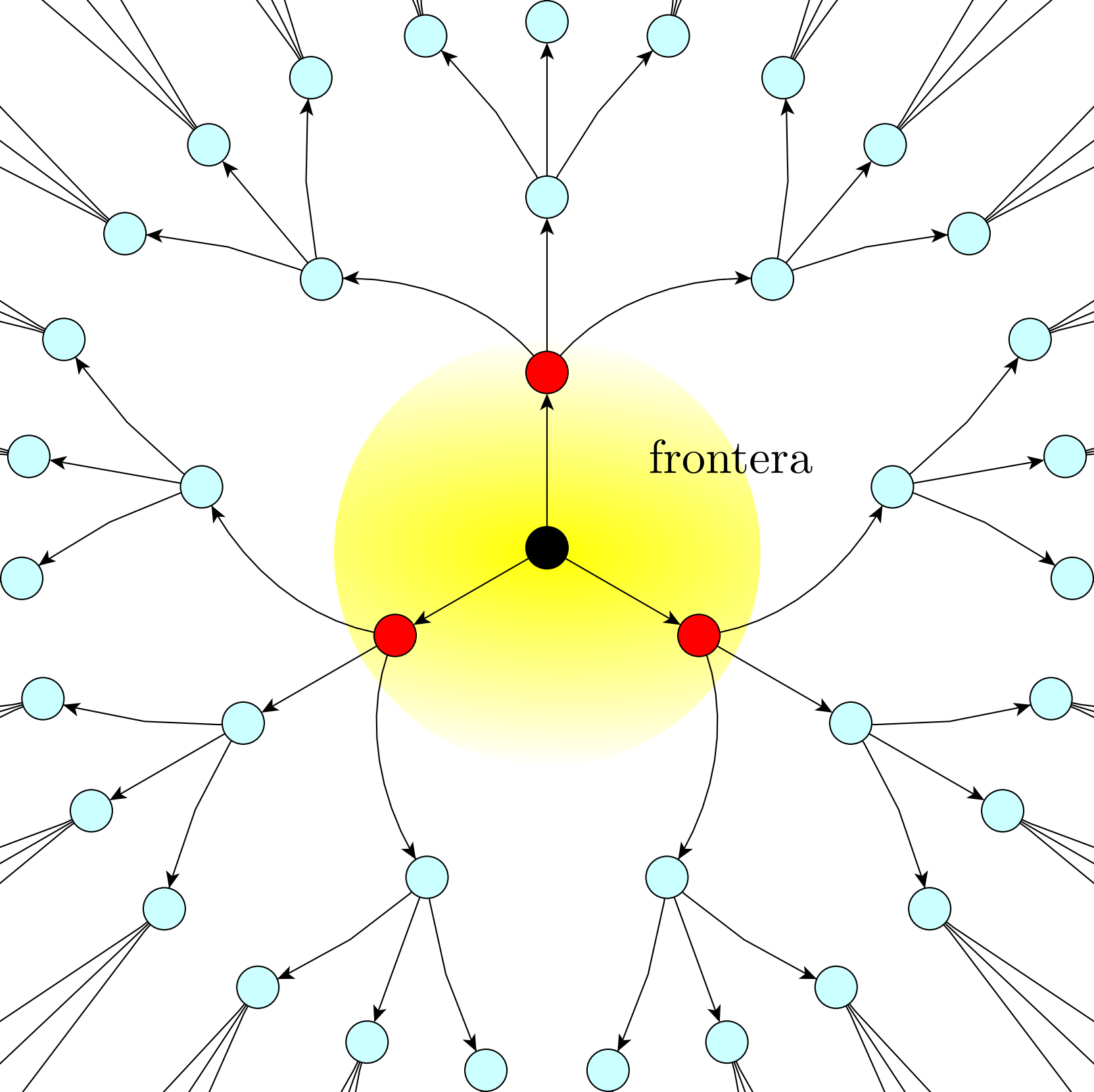
- Profundidad de la solución (d).

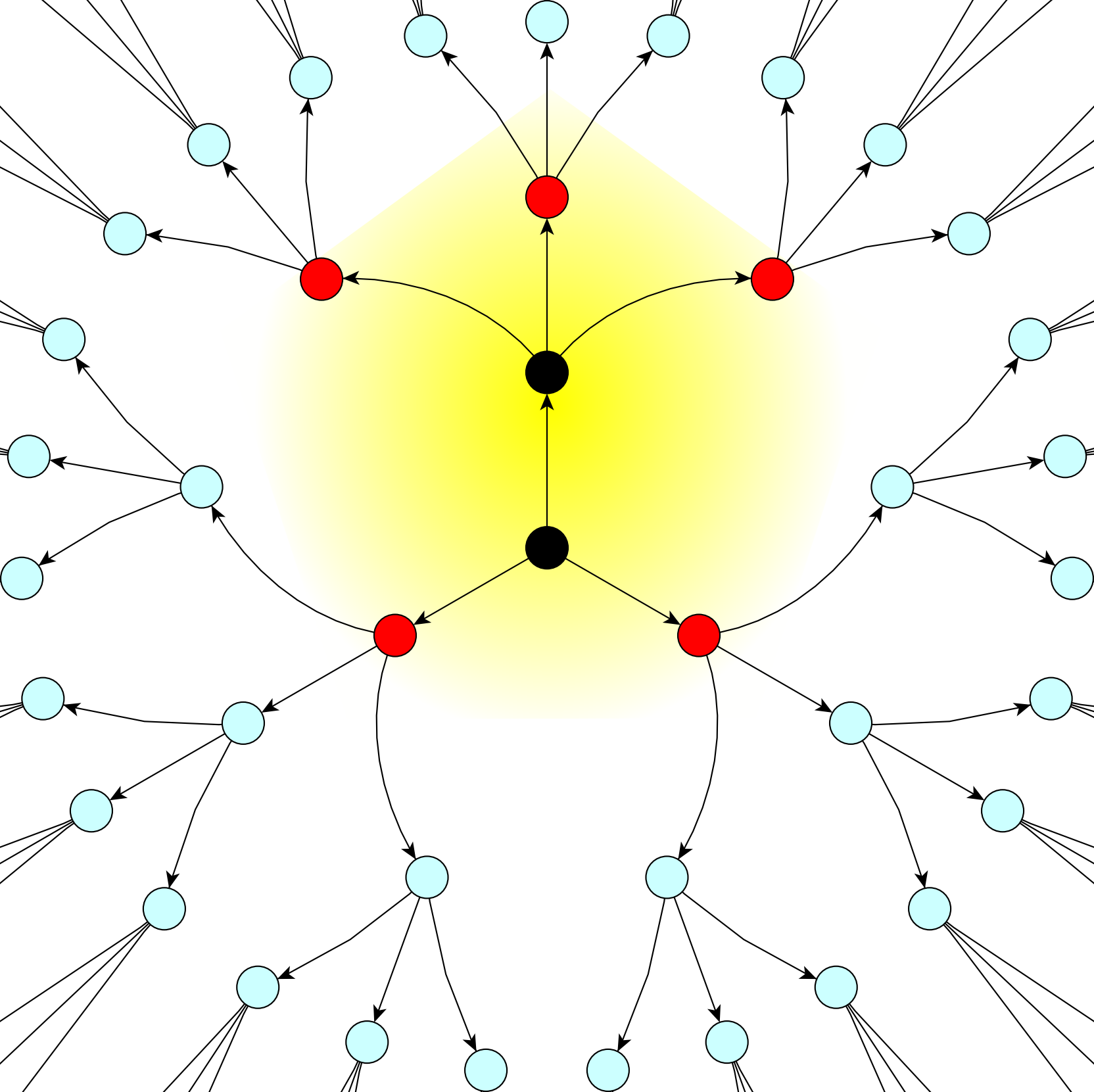
Distancia del estado inicial a la solución más cercana.

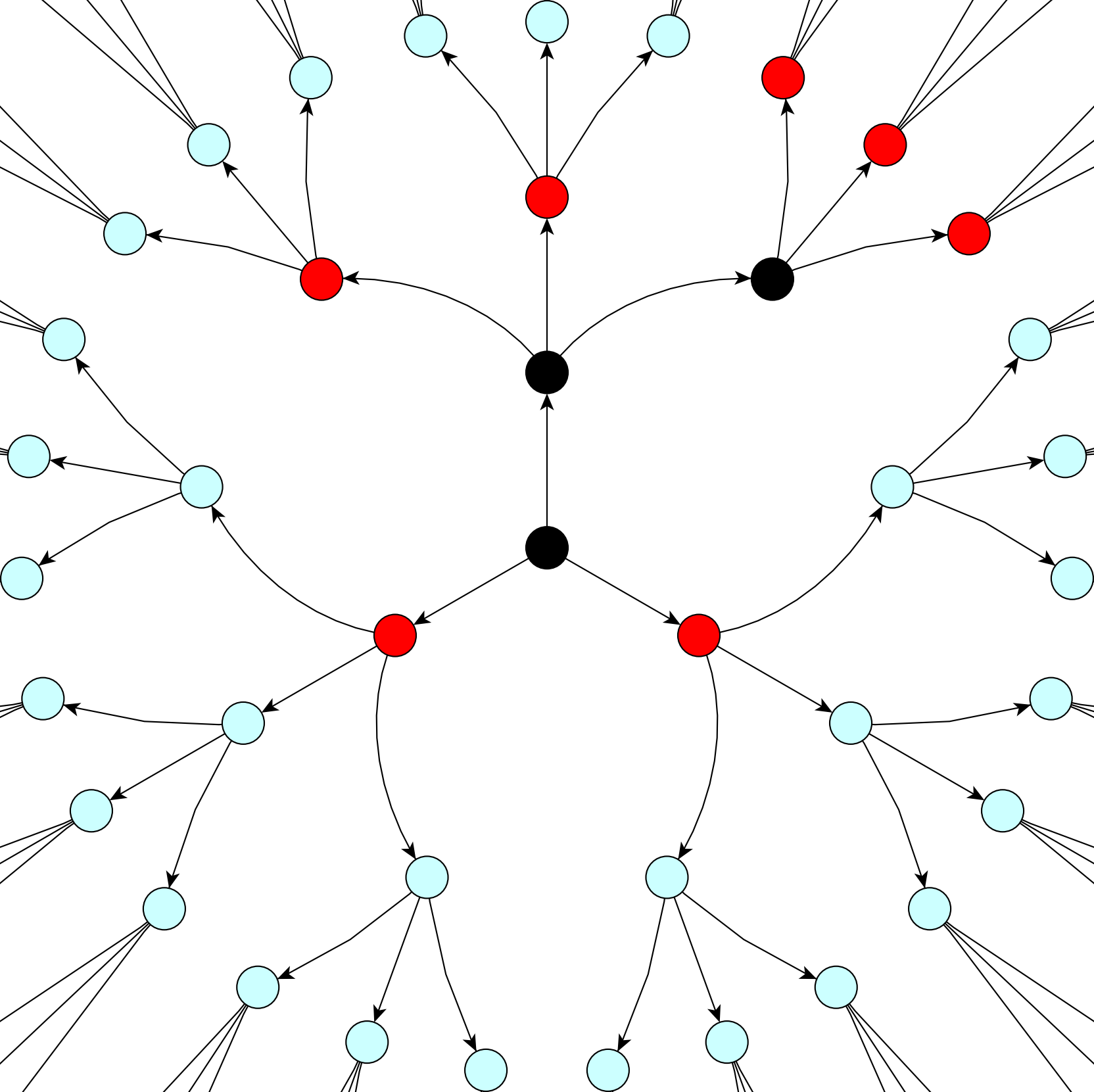


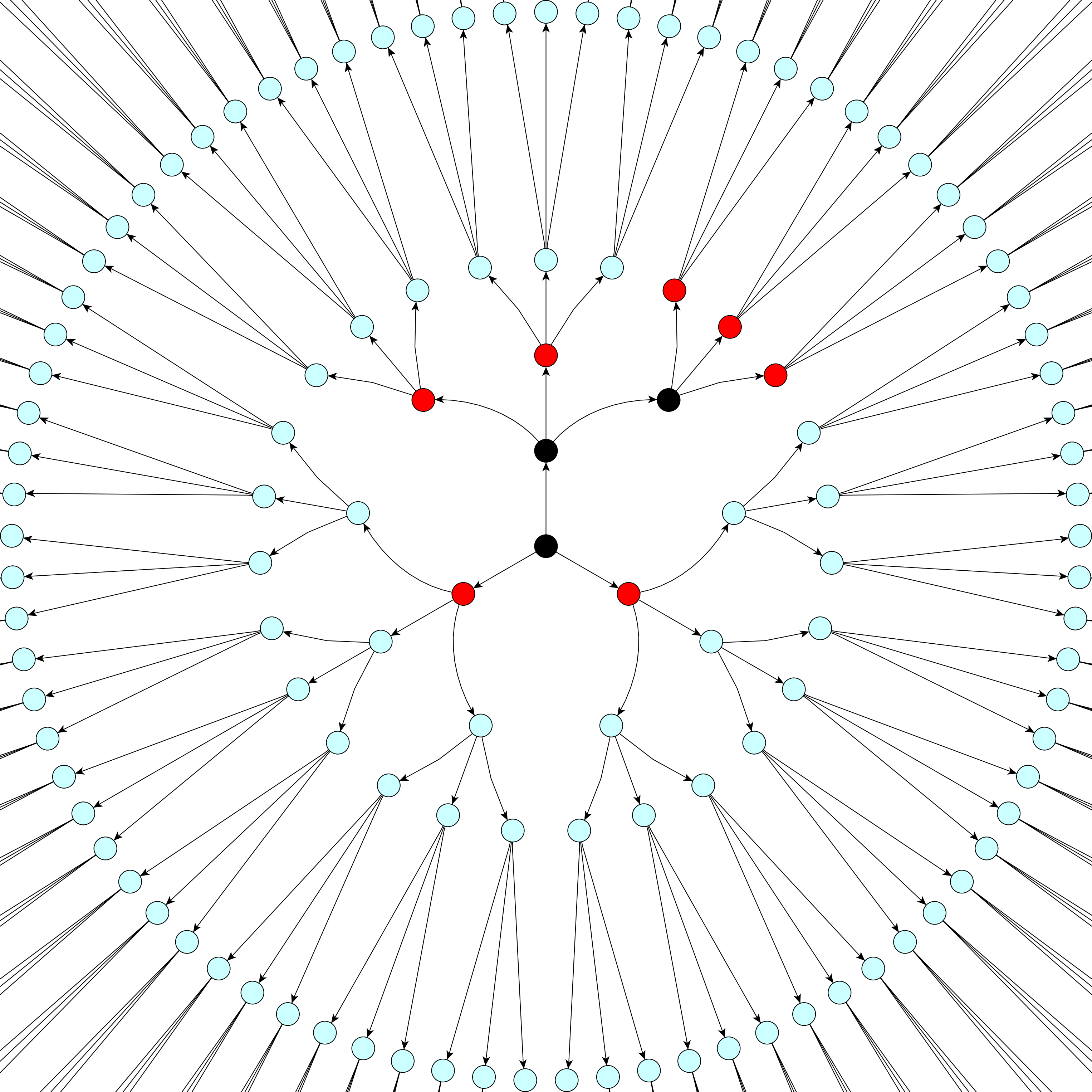


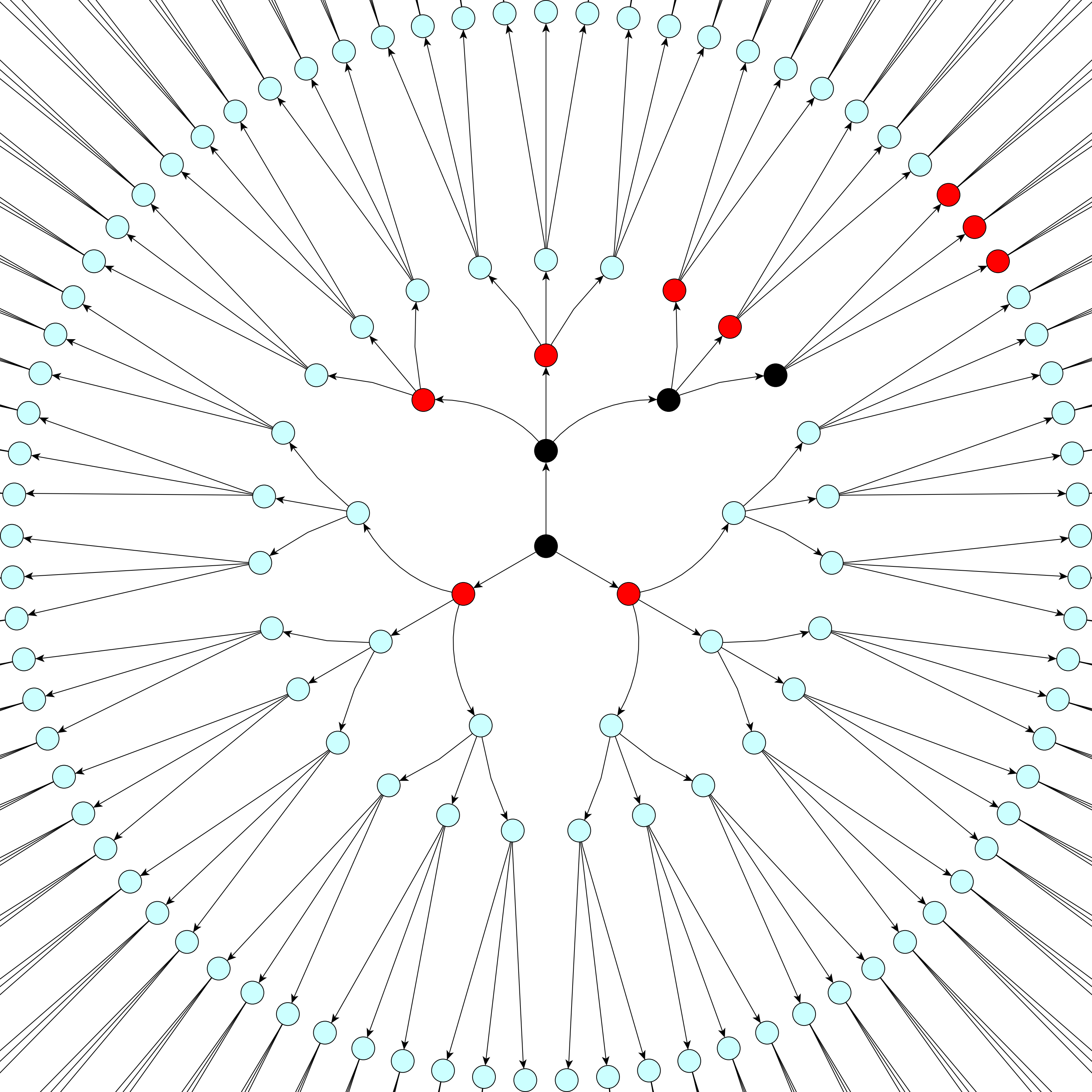


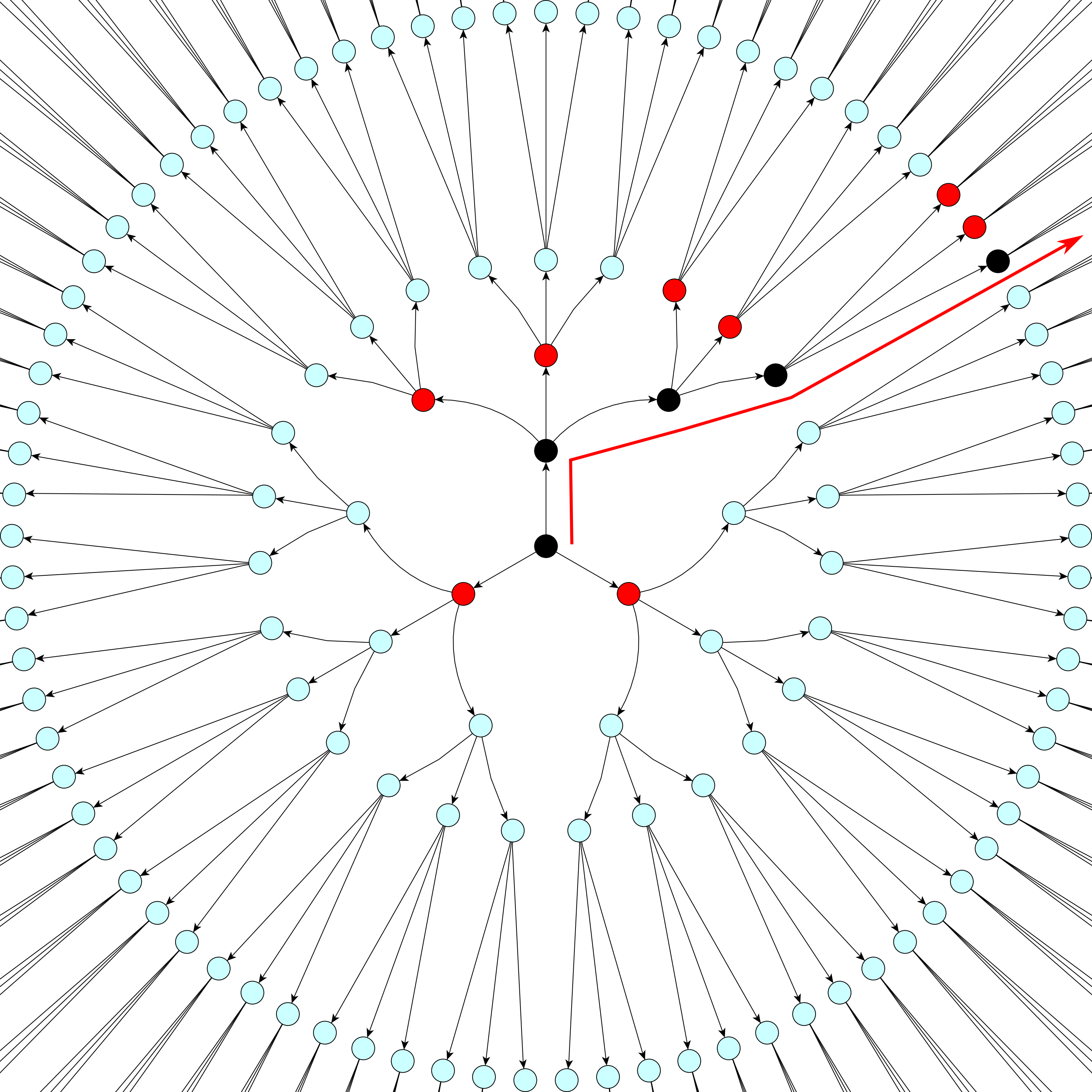


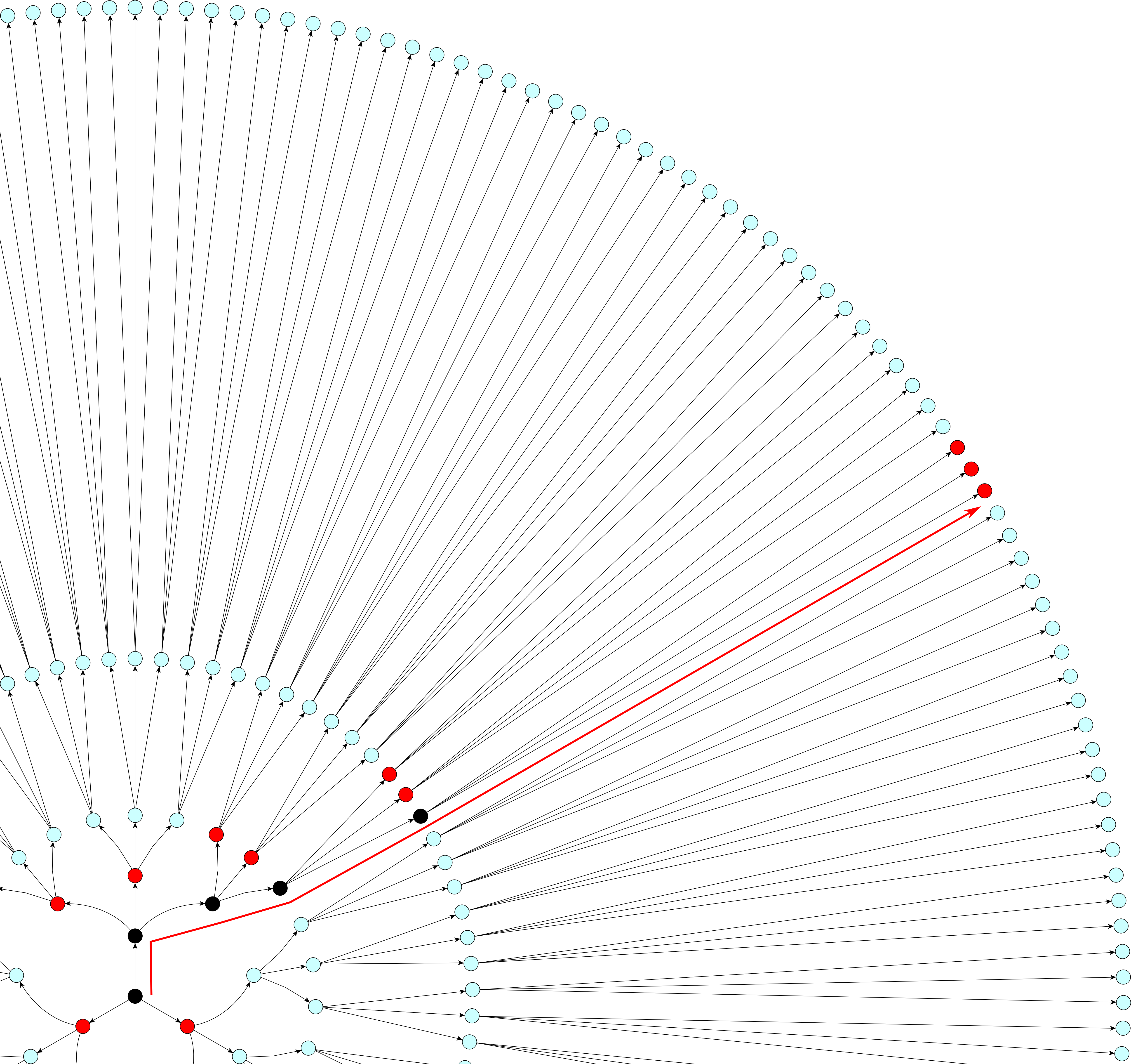


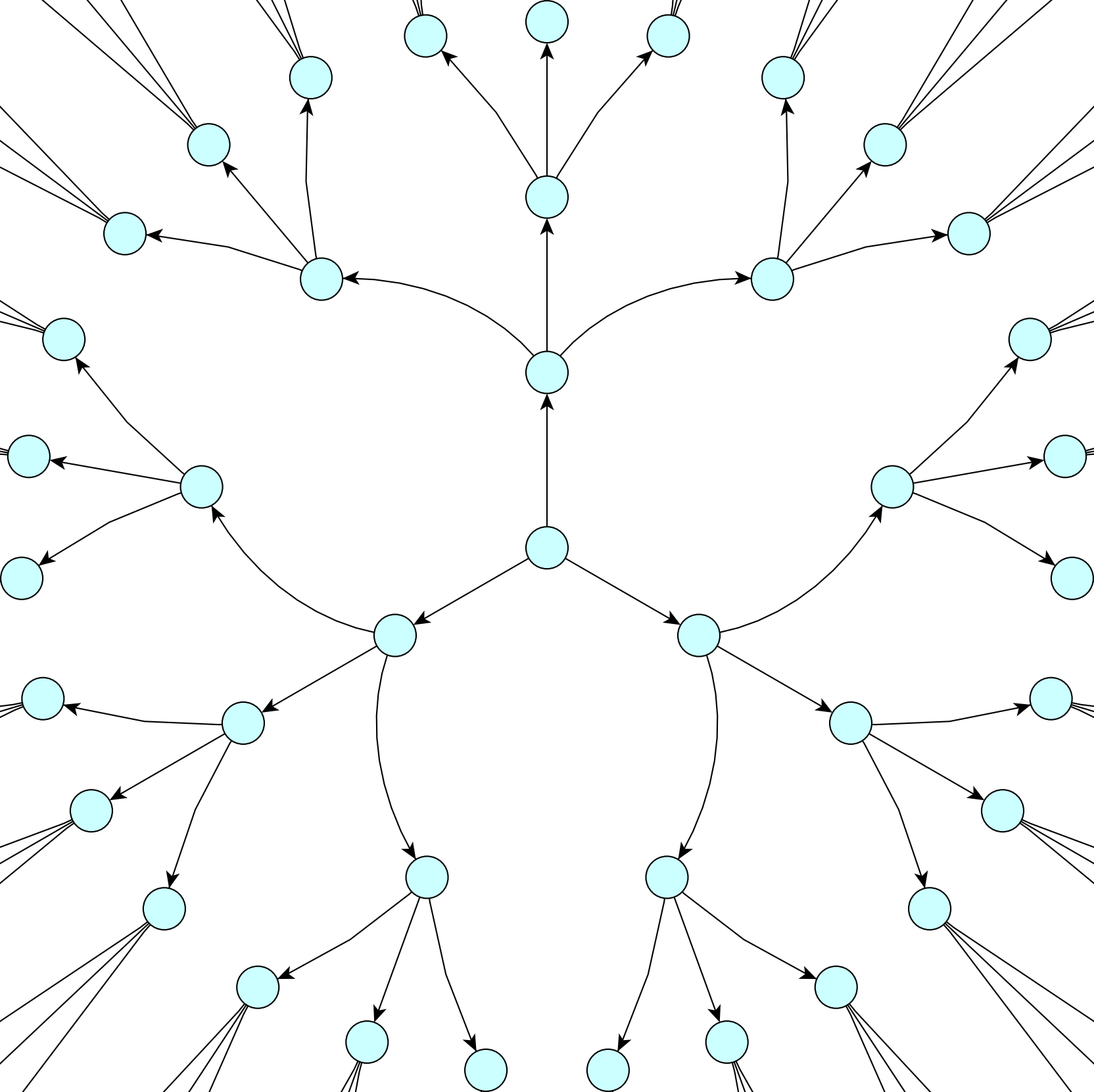


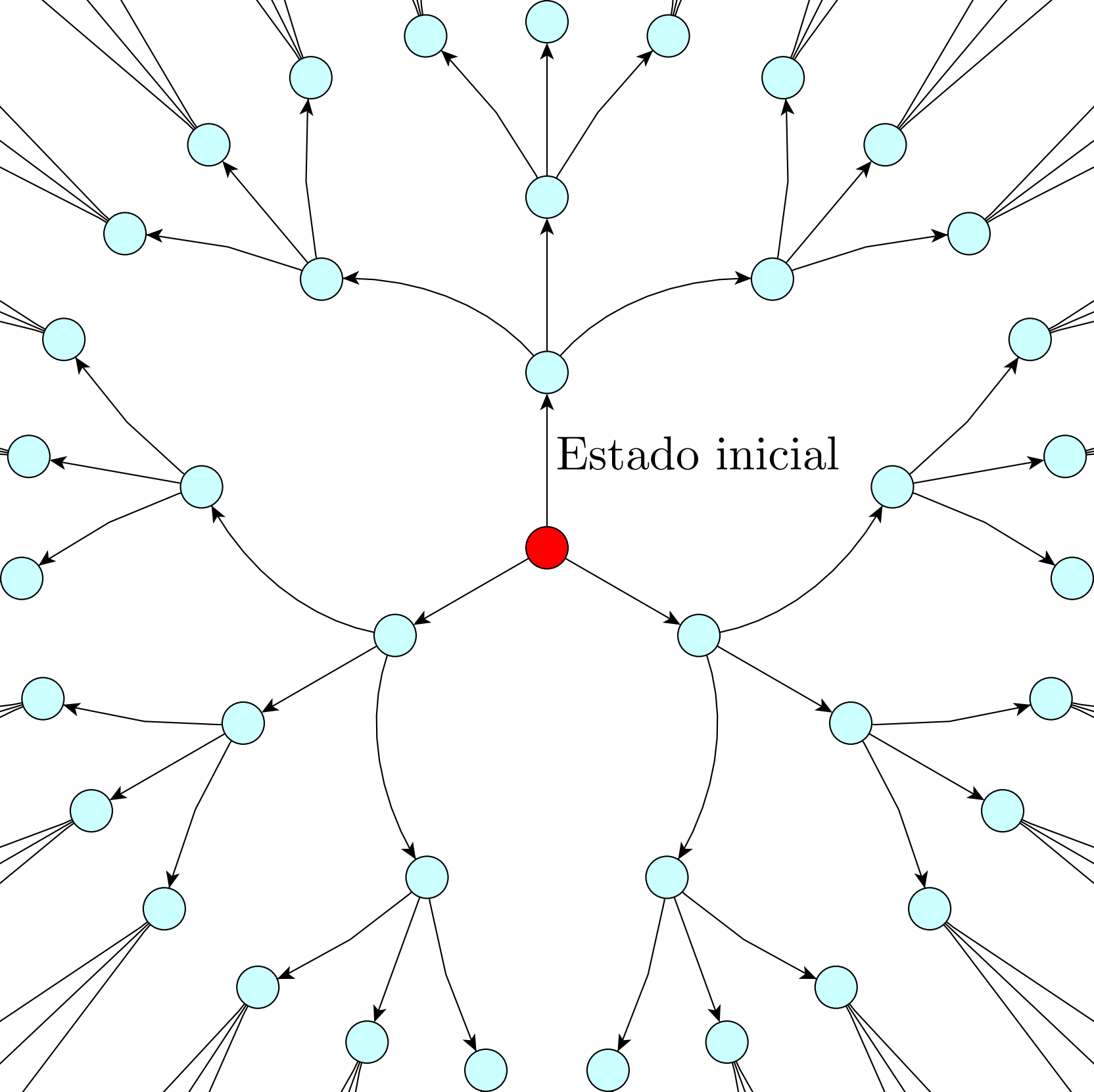


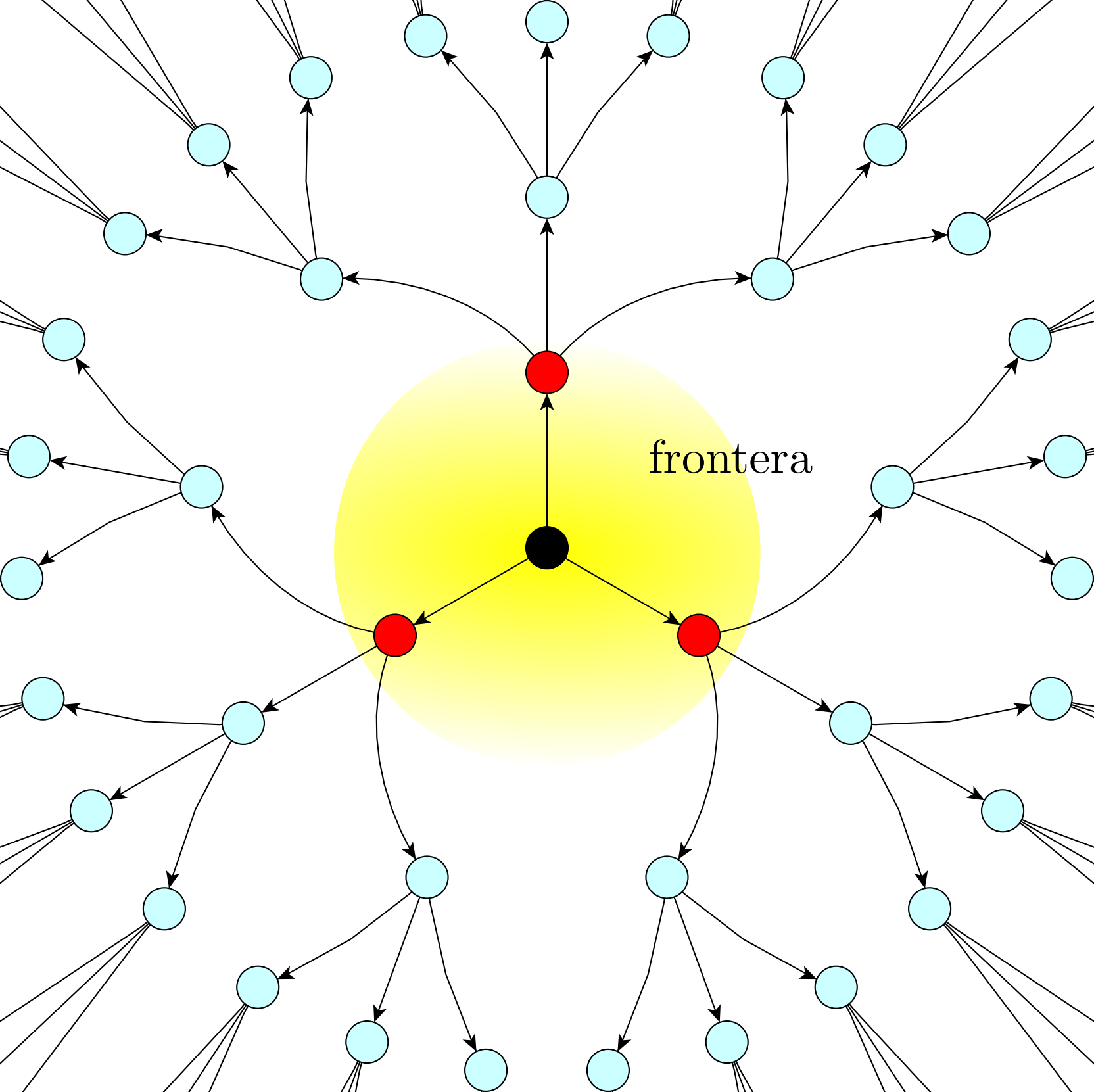


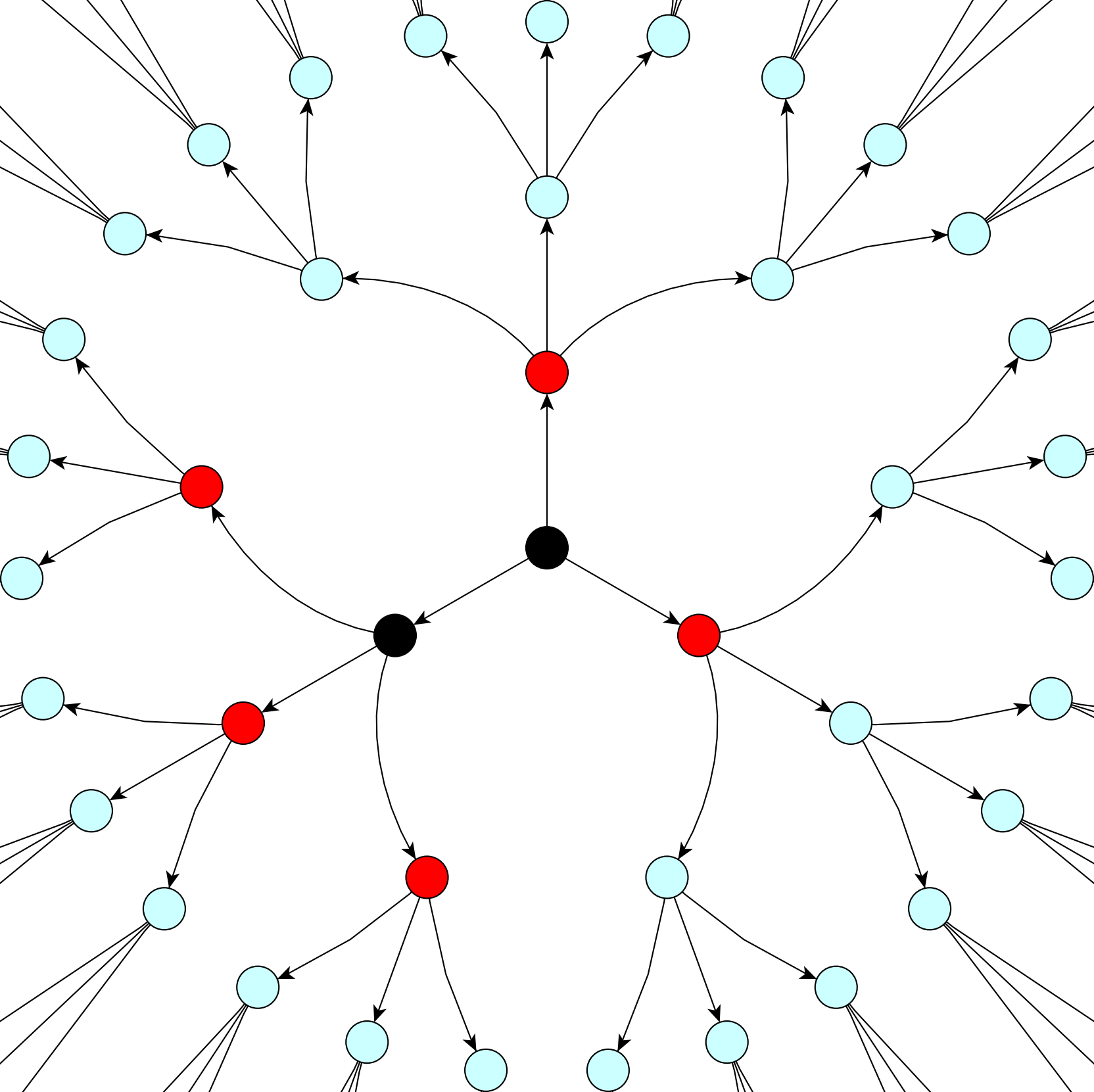


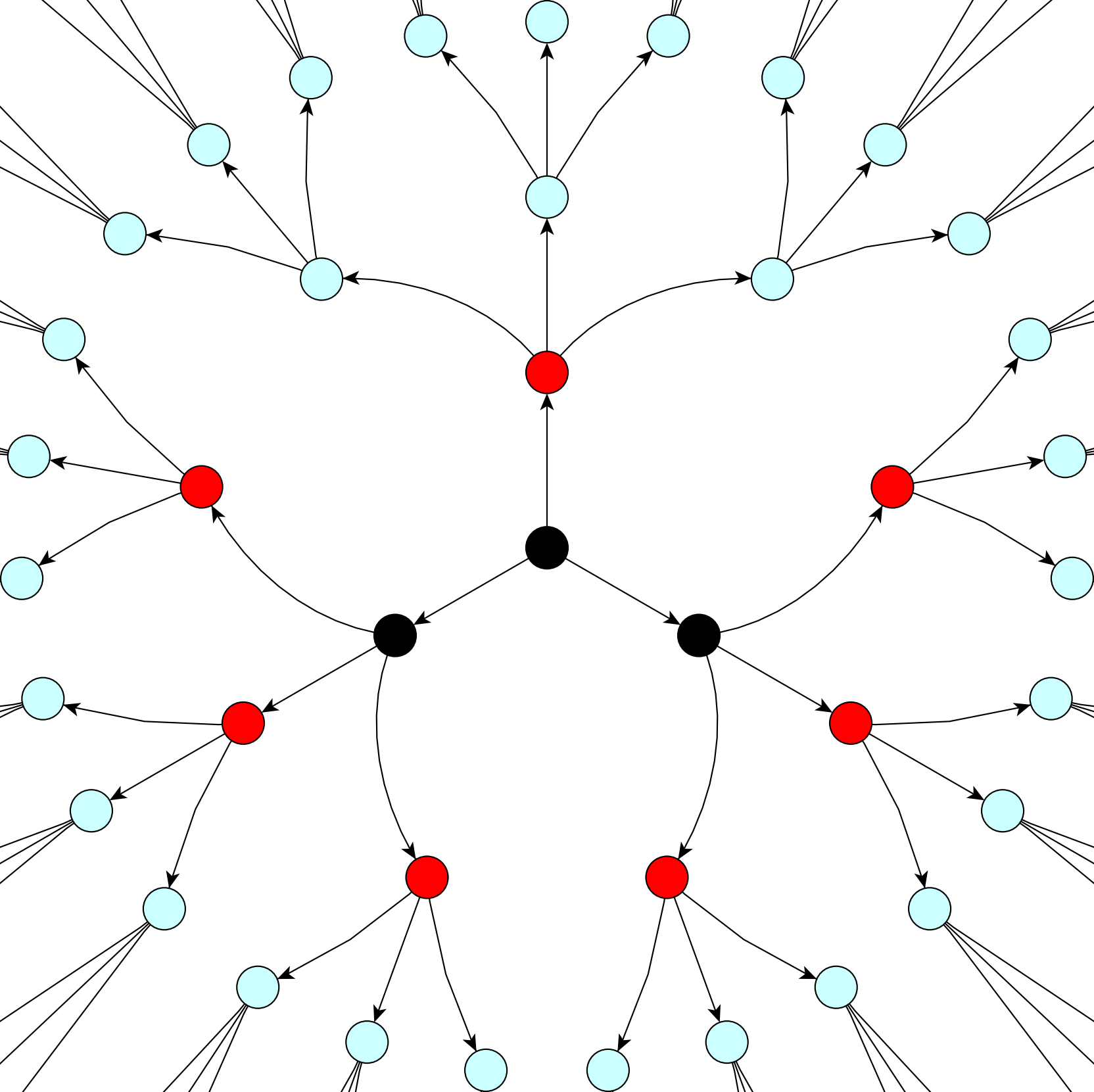


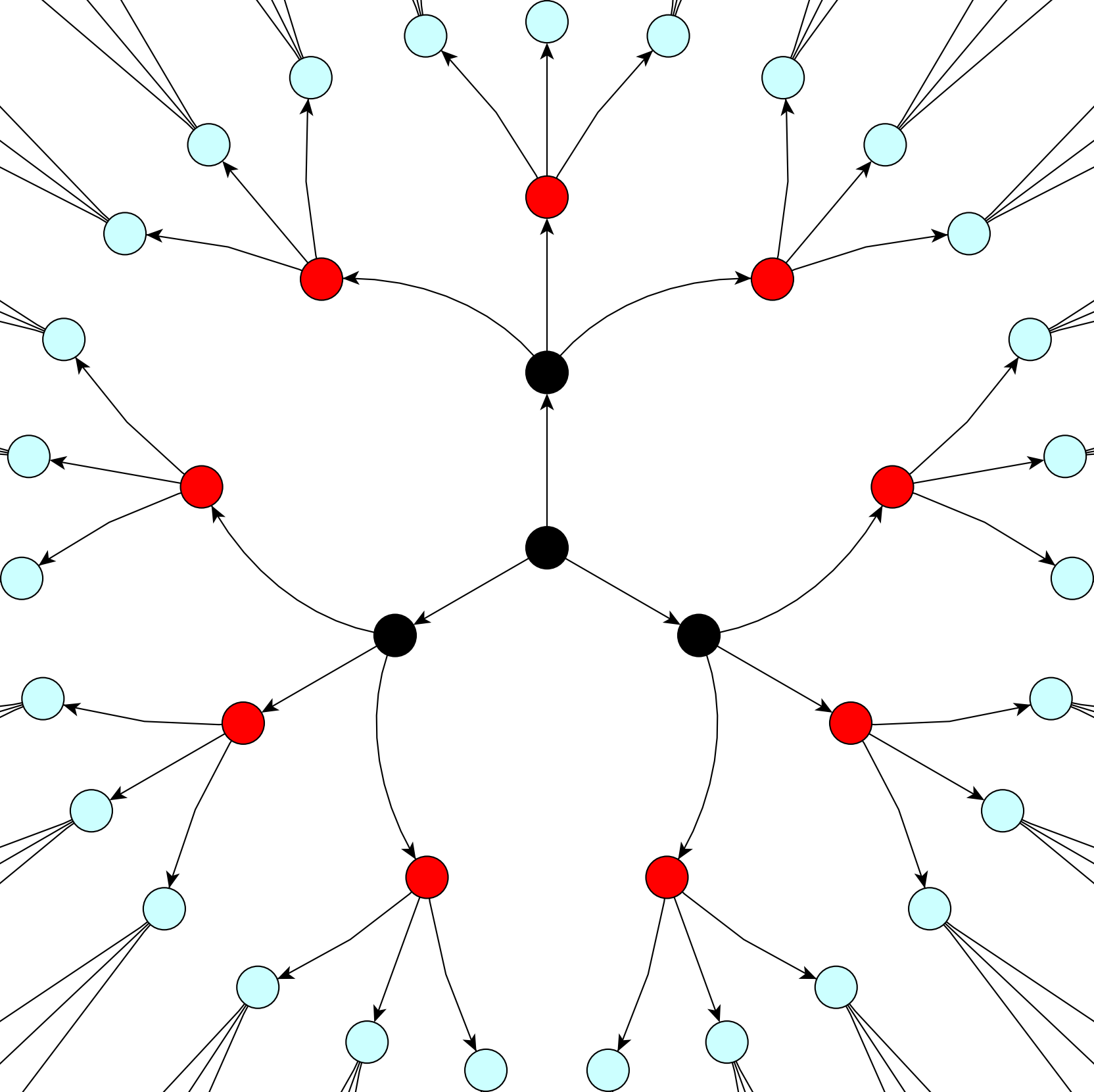


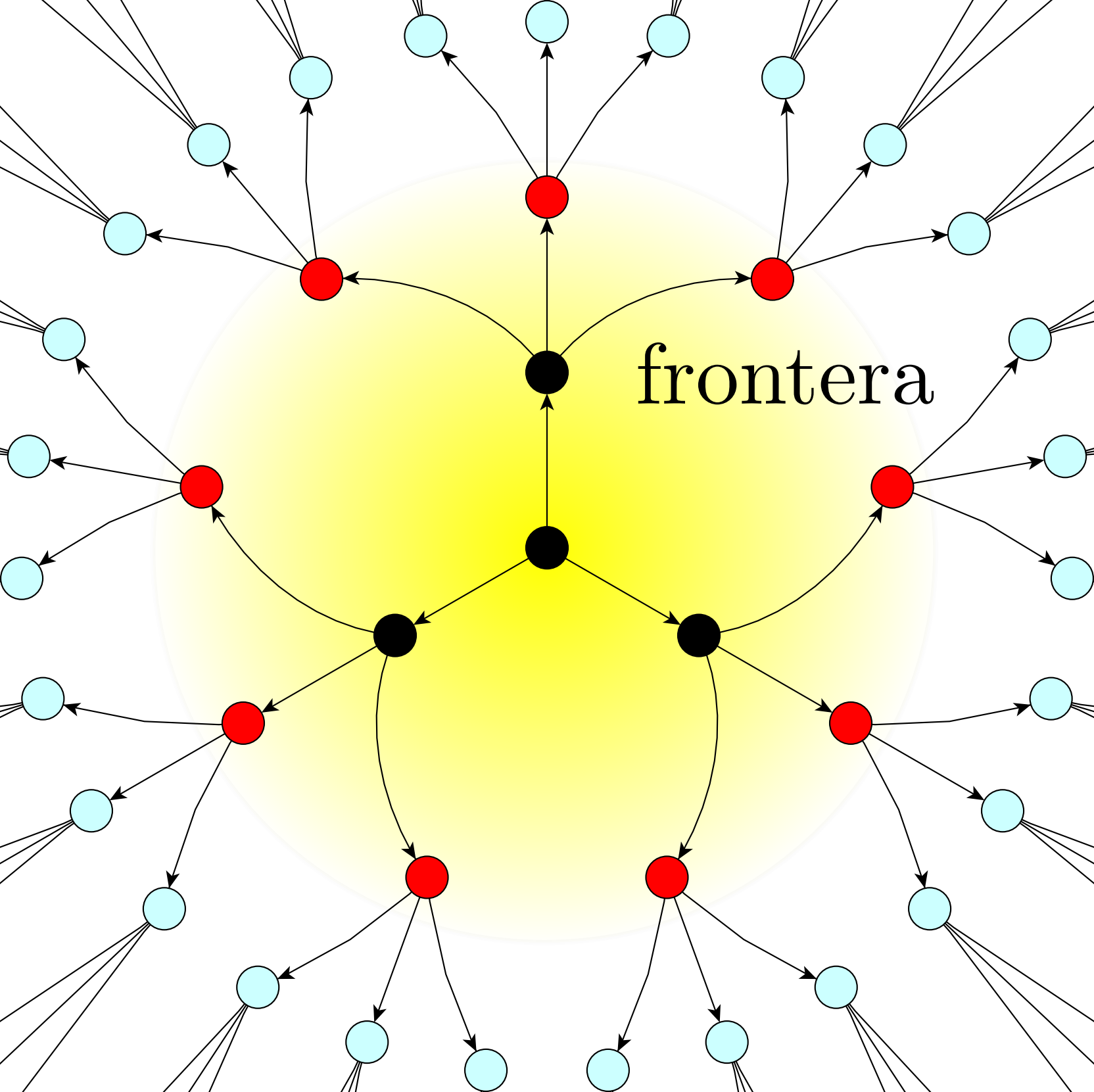


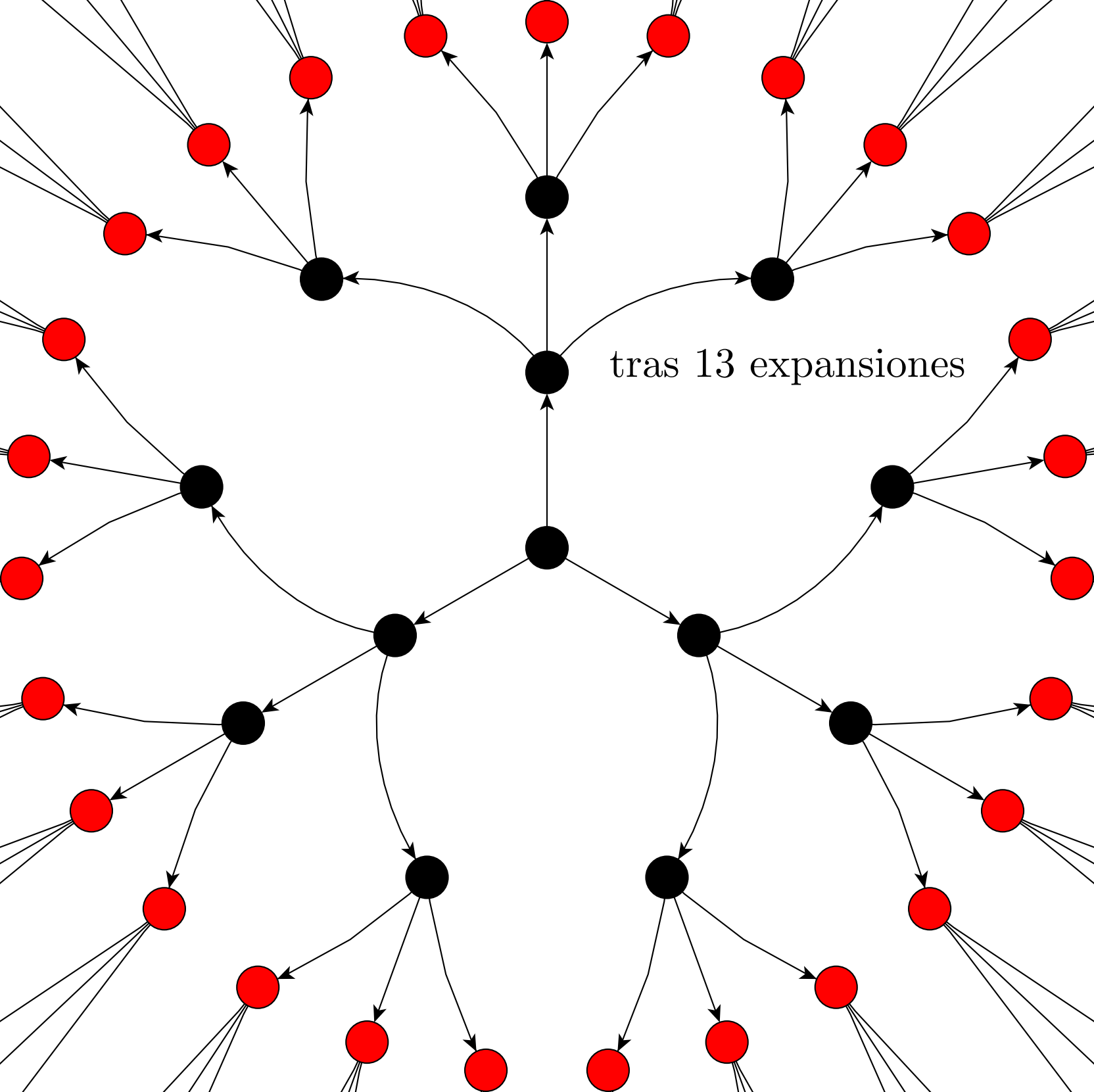


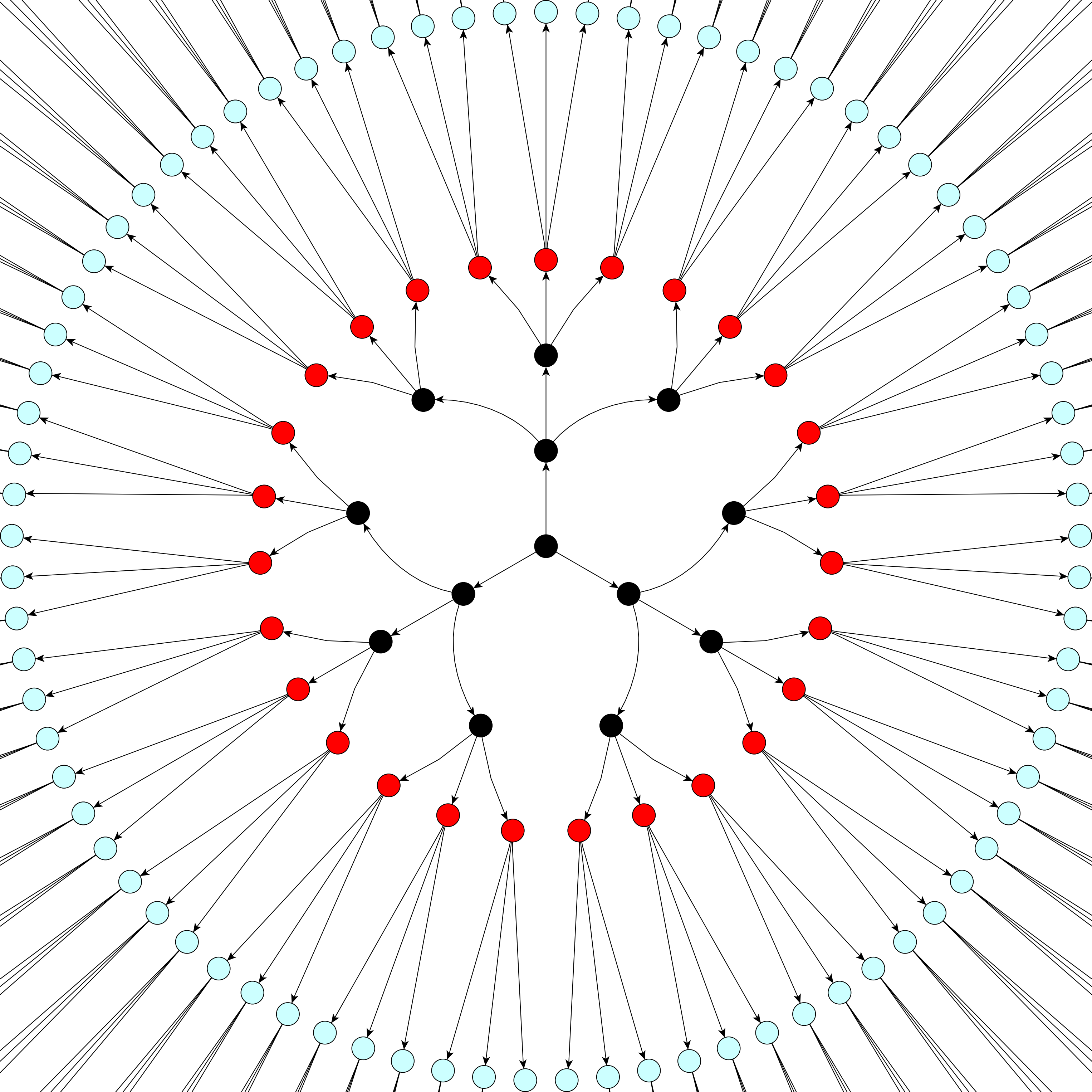


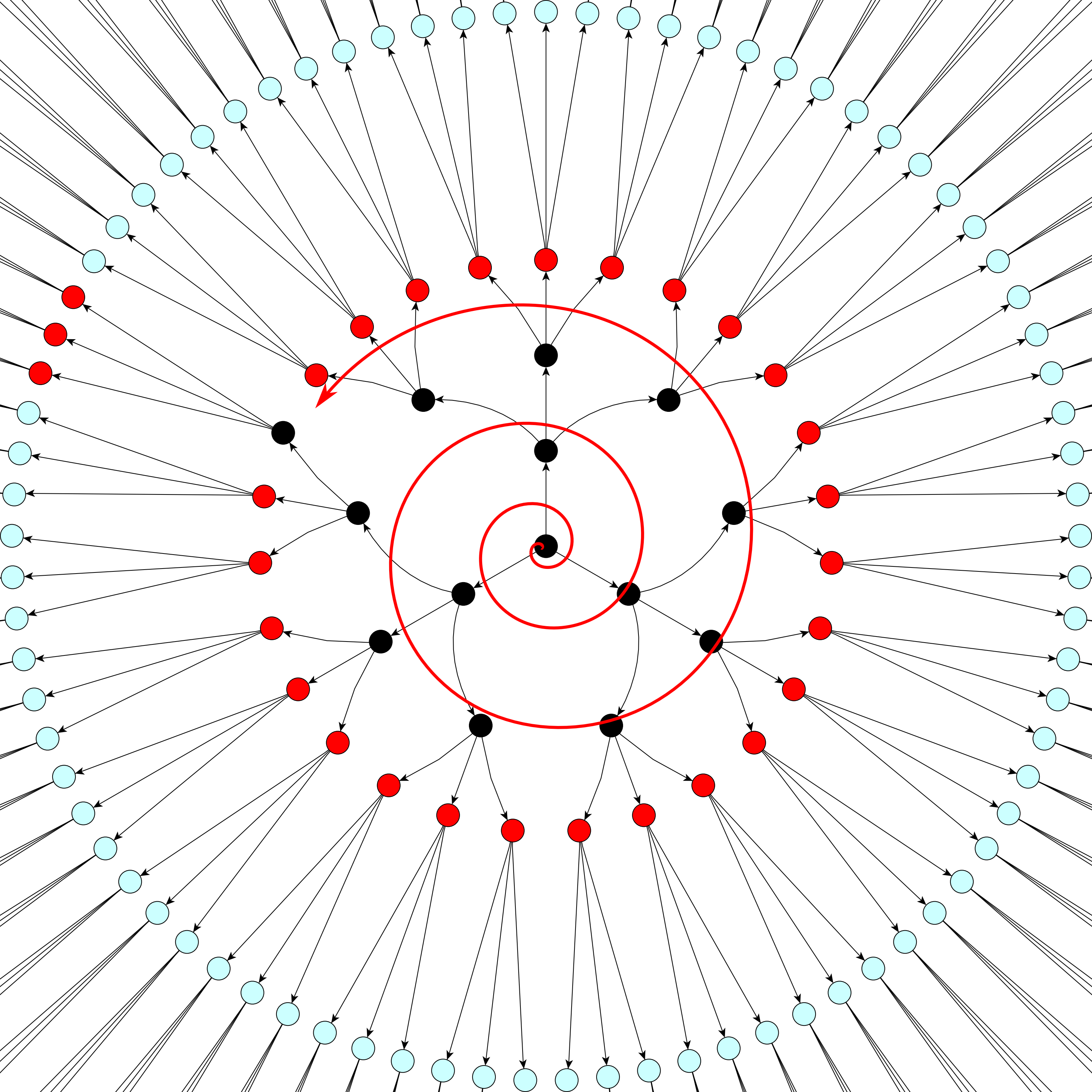


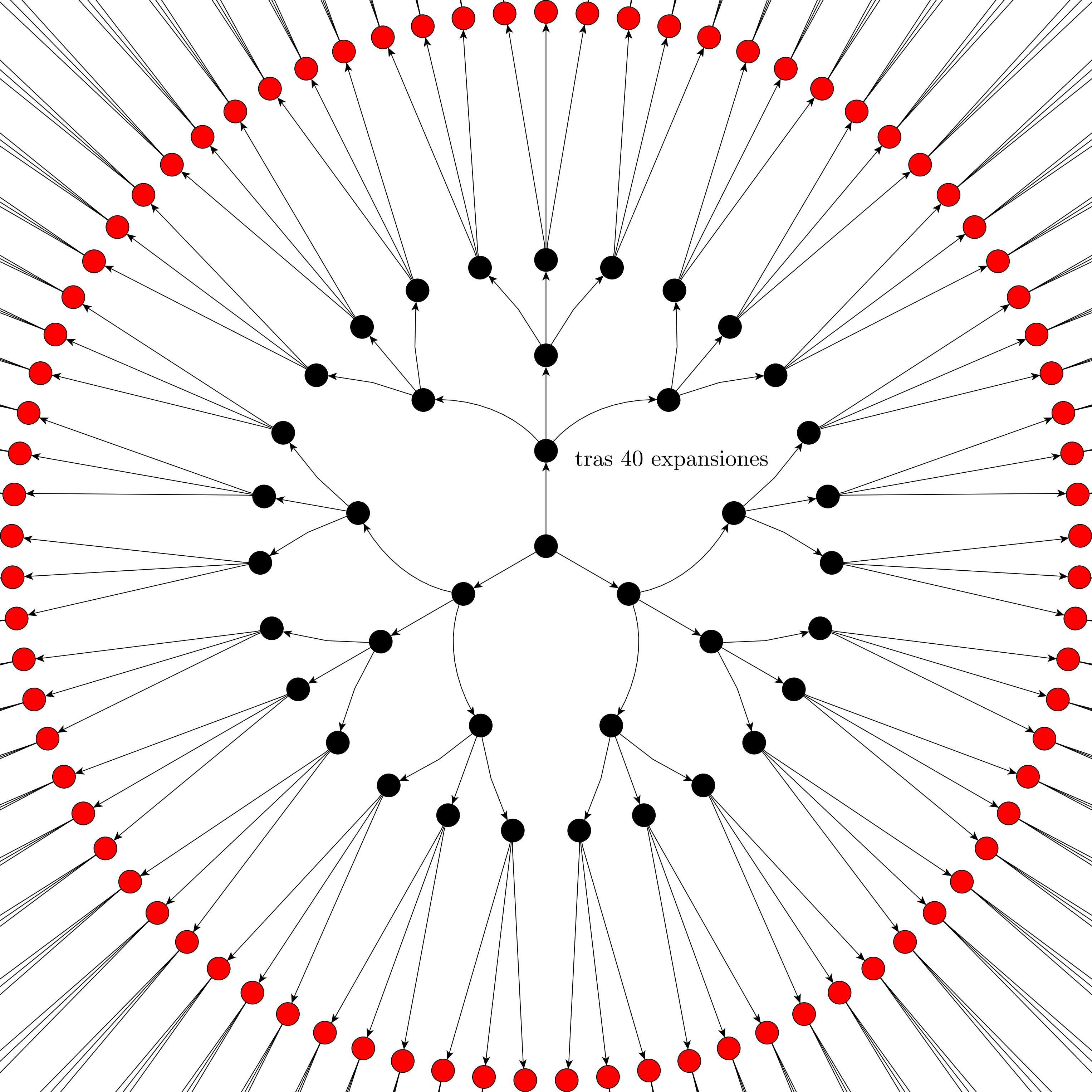


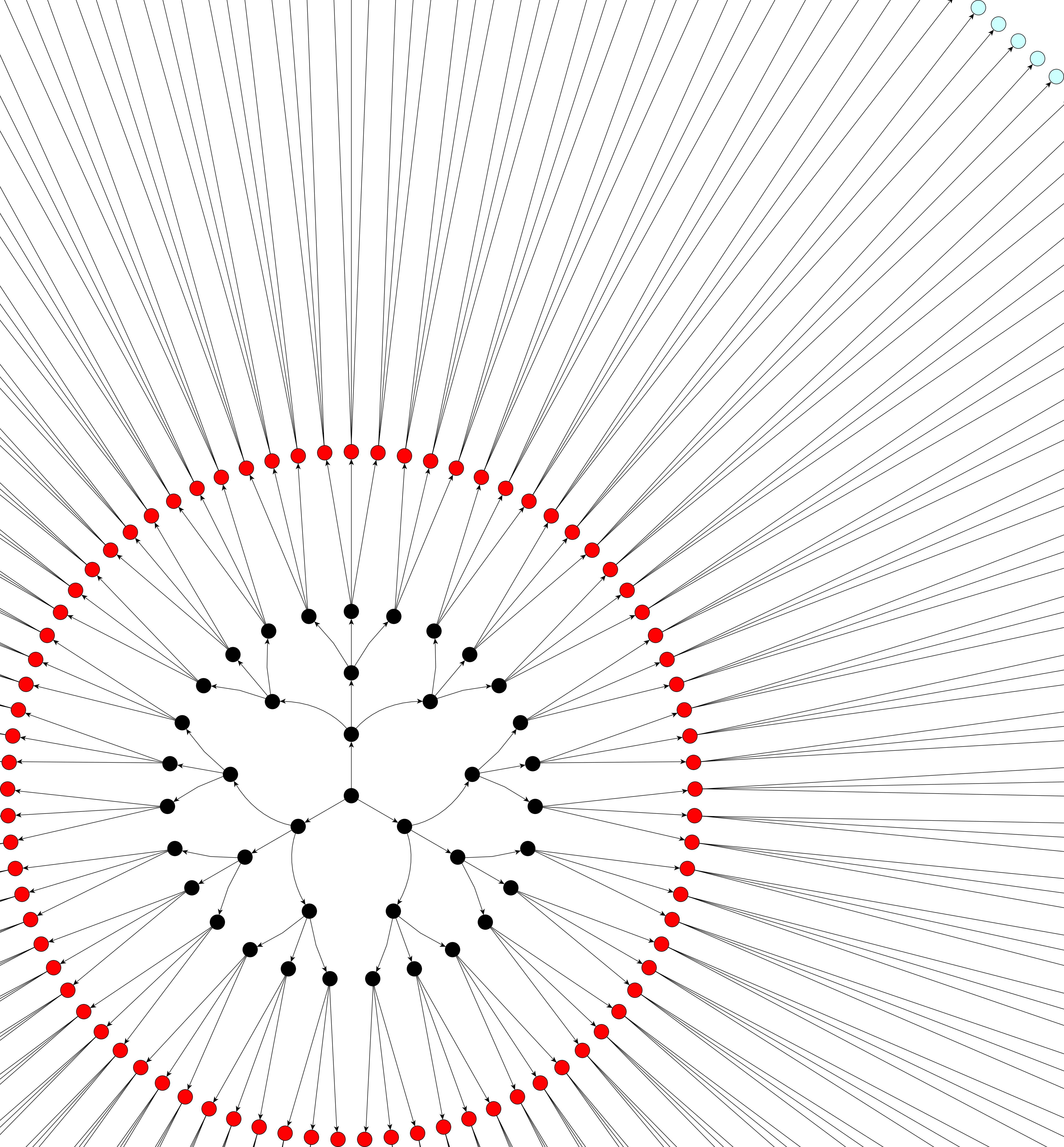


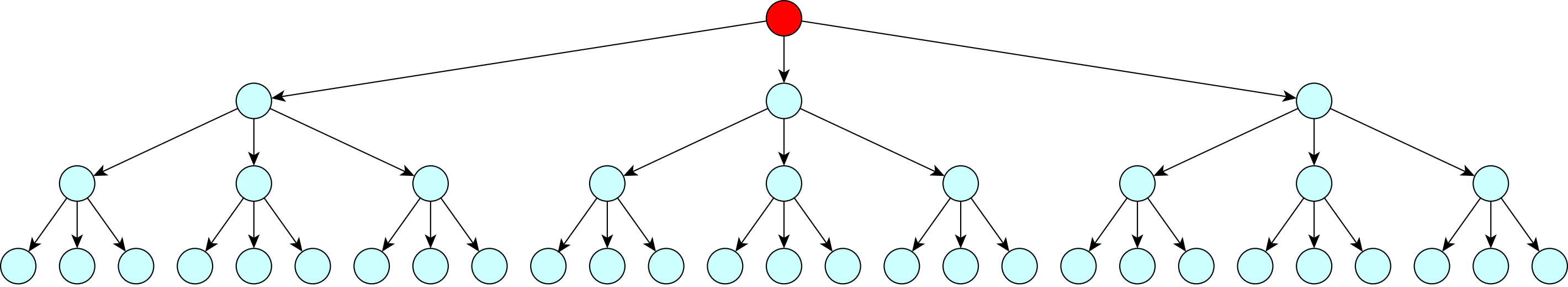










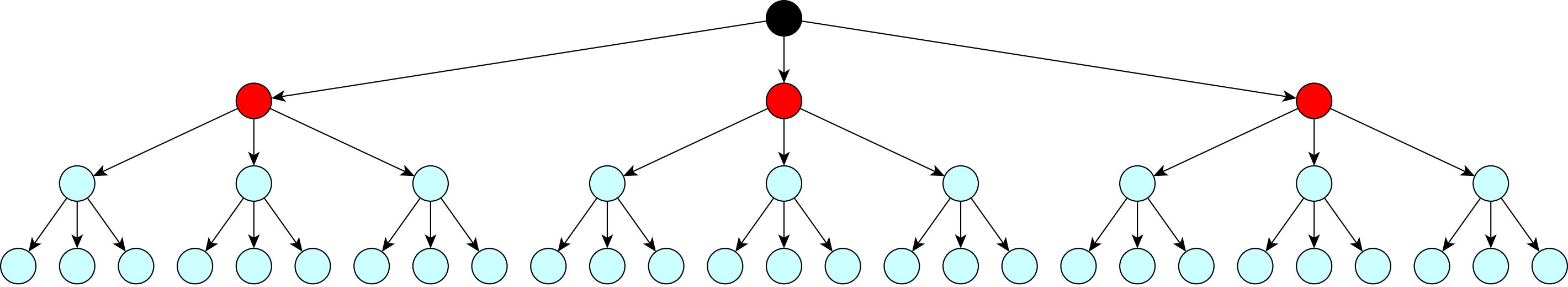


Algoritmo DFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0

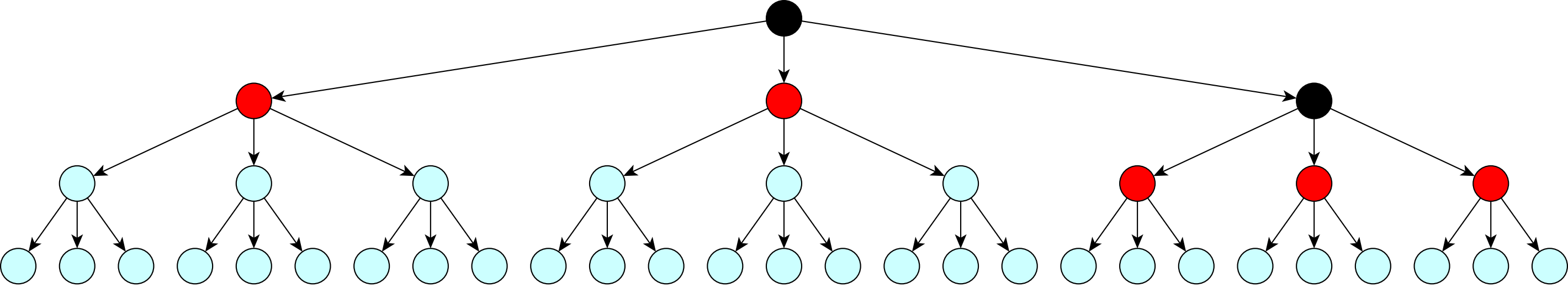


Algoritmo DFS

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.

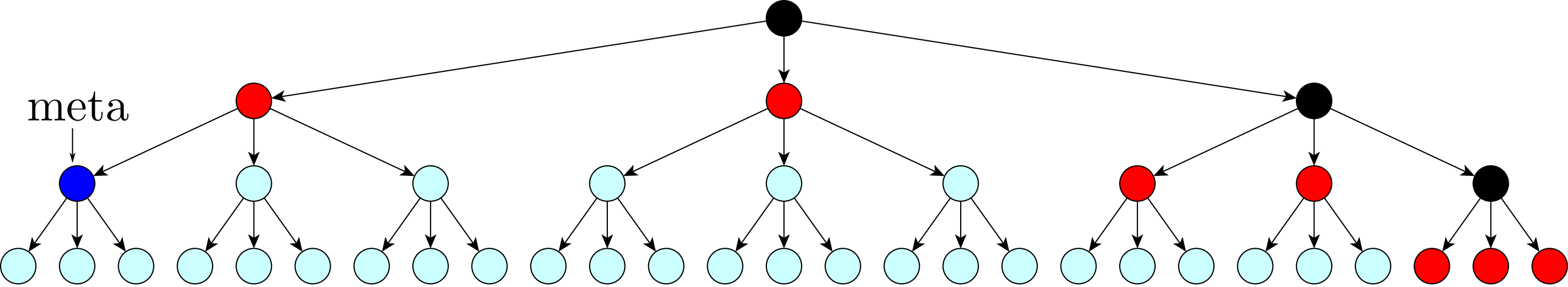
Observaremos como crece la frontera
y conjunto de expandidos

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



- $O(bn)$

2. Tiempo.



Algoritmo DFS

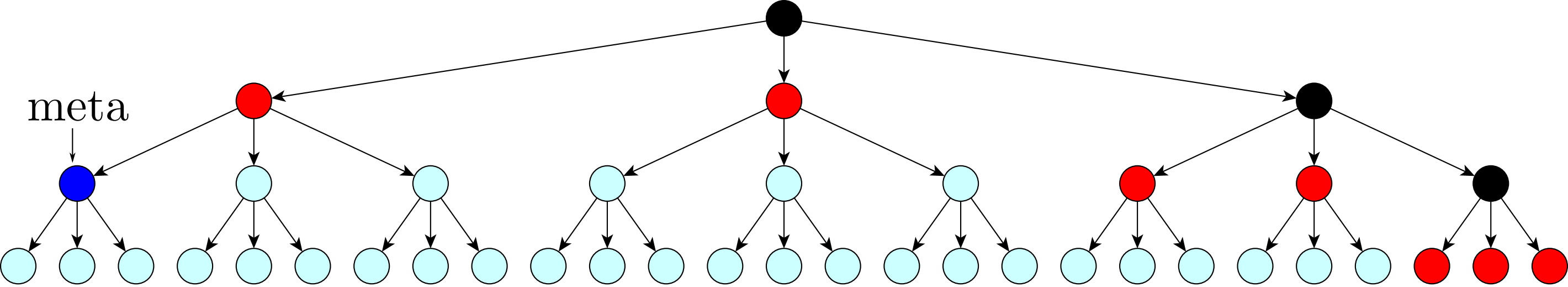
1. Memoria.

■ $O(bn)$

2. Tiempo.

Peor escenario:

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



Algoritmo DFS

1. Memoria.

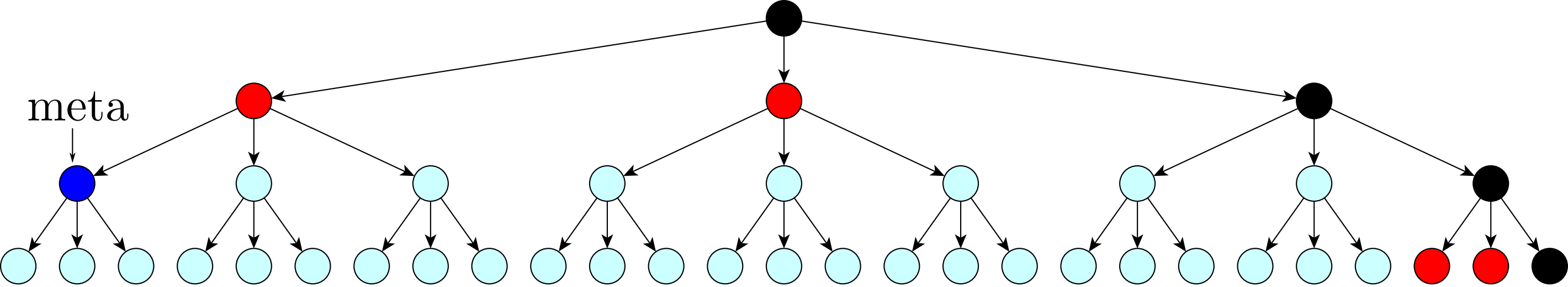
■ $O(bn)$

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.

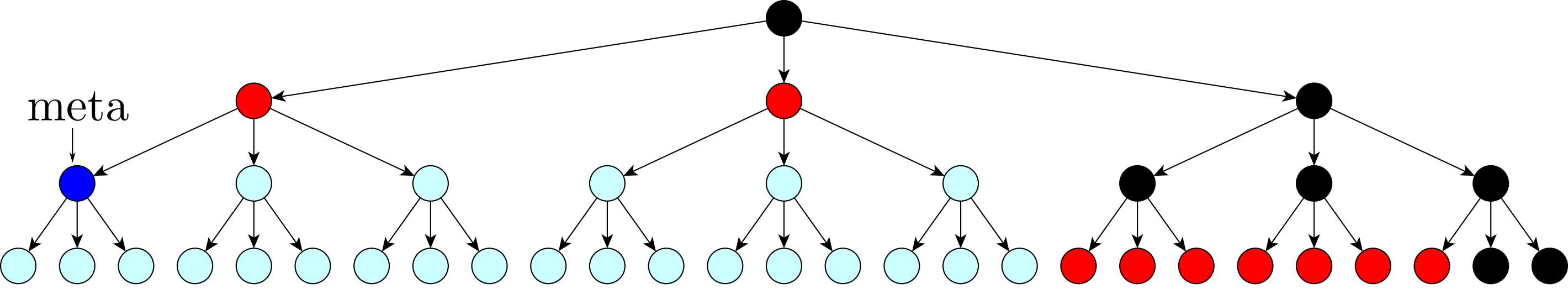
- $O(bn)$

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.

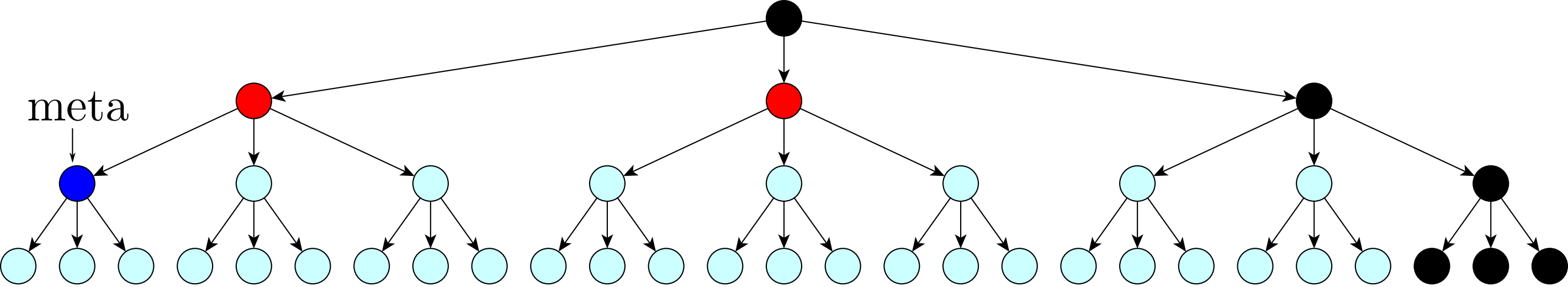
- $O(bn)$

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.

■ $O(bn)$

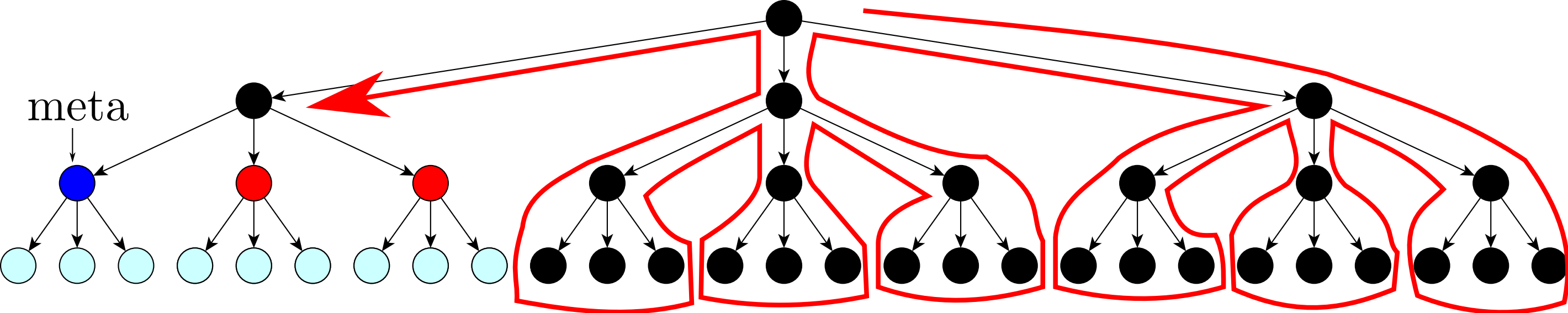
2. Tiempo.

Peor escenario:

La meta esta 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.

$O(b^m)$

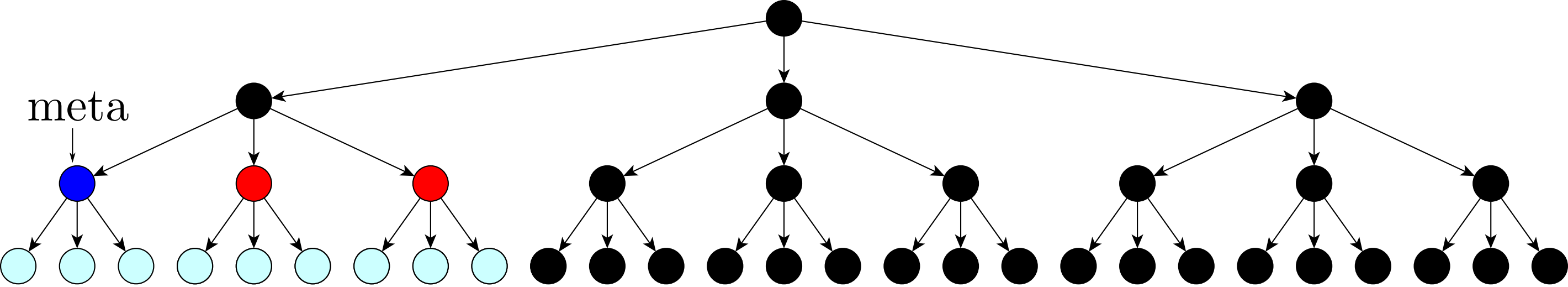
Peor escenario:

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.

- $O(b^m)$

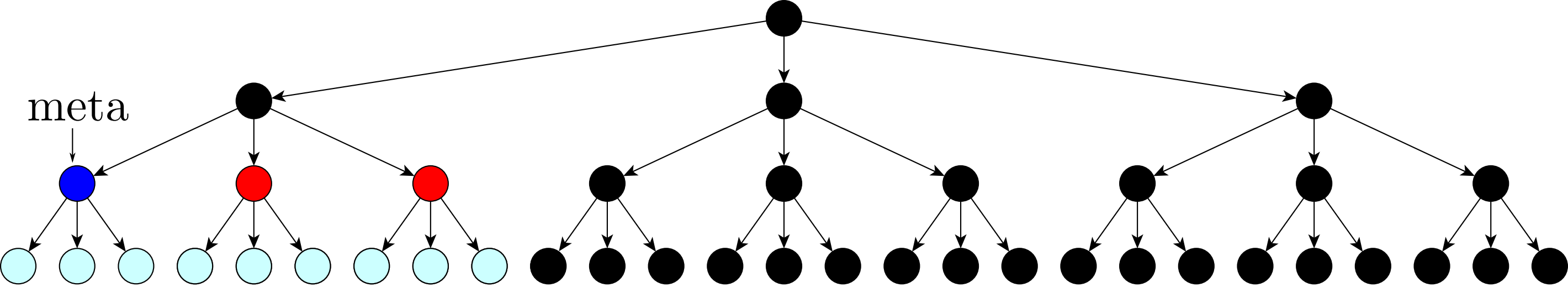
3. Calidad.

- 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.

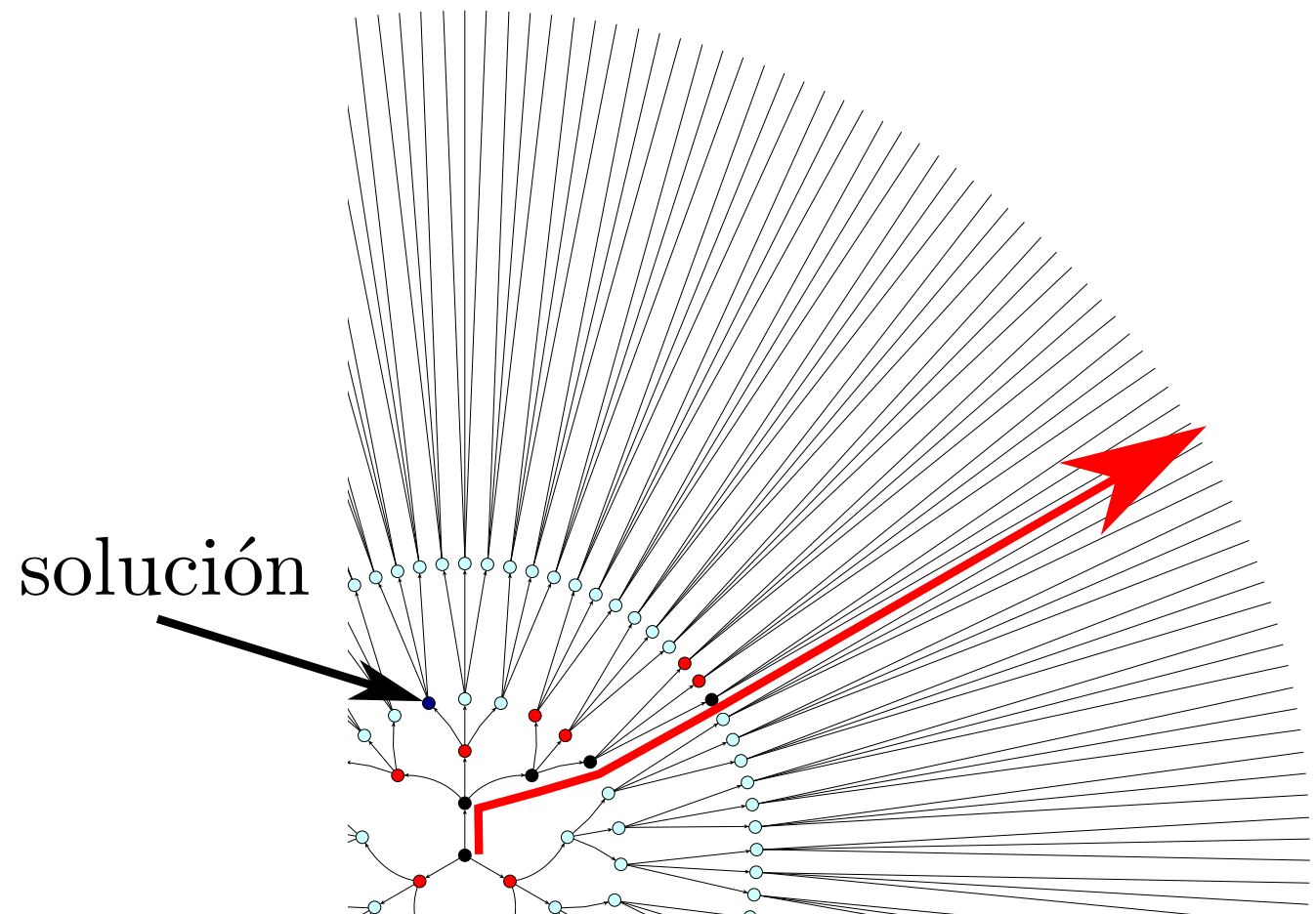
- $O(b^m)$

3. Calidad.

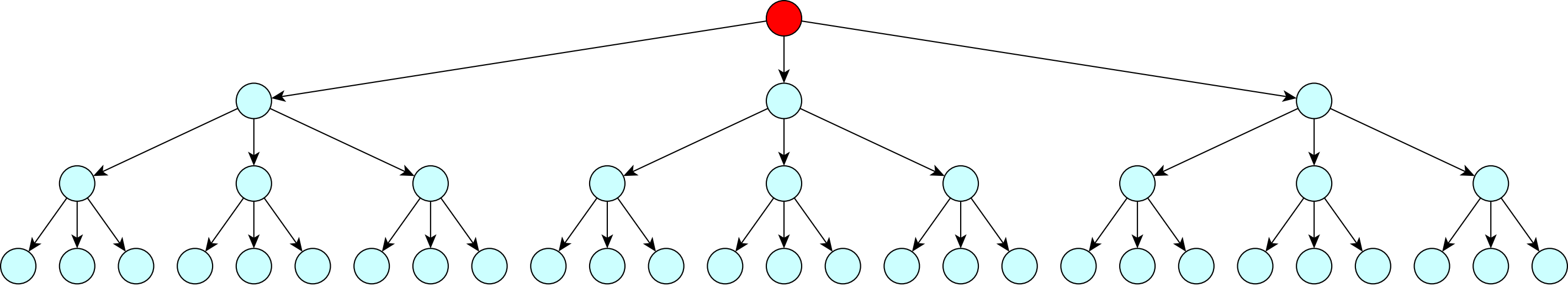
- Solución no óptima.

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.

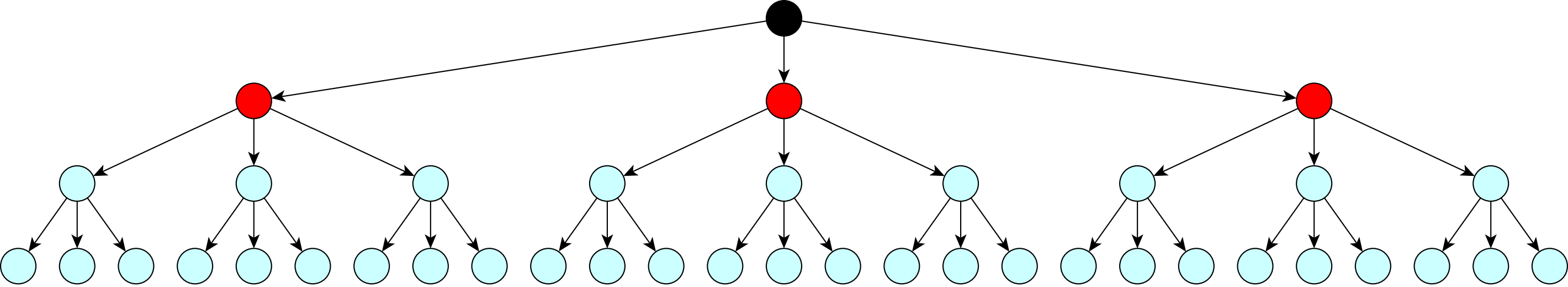


Algoritmo BFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0

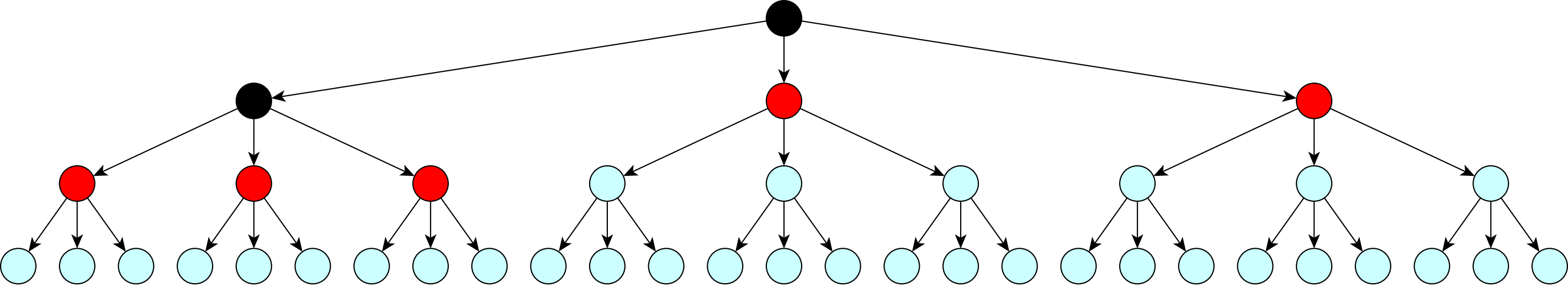


Algoritmo BFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

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

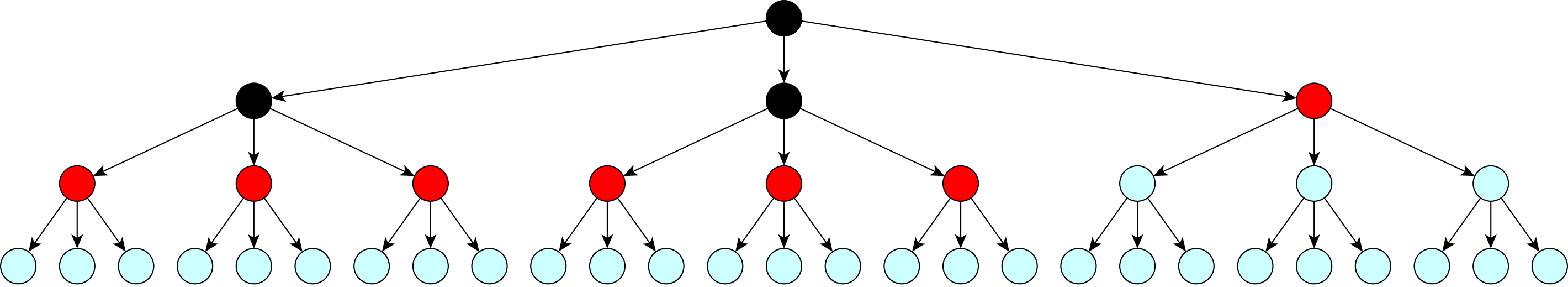


Algoritmo BFS

Observaremos como crece la frontera
y conjunto de expandidos

1. Memoria.

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

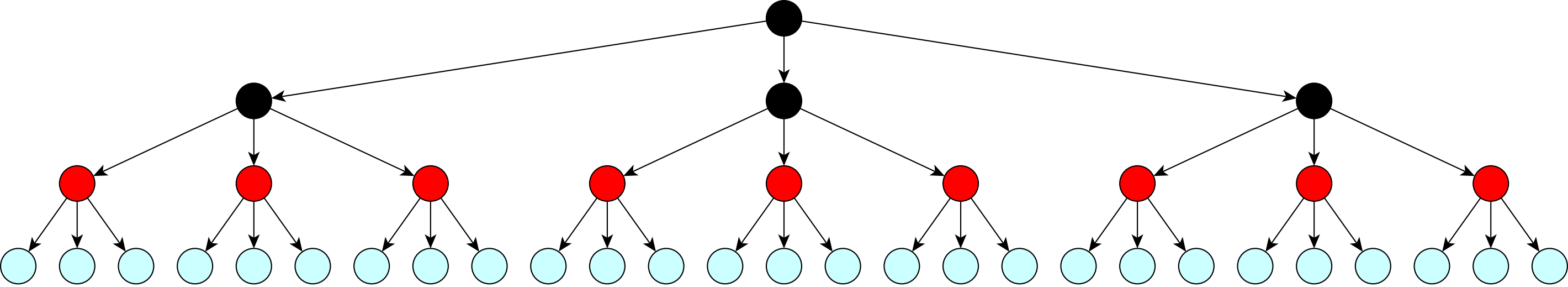


Algoritmo BFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

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

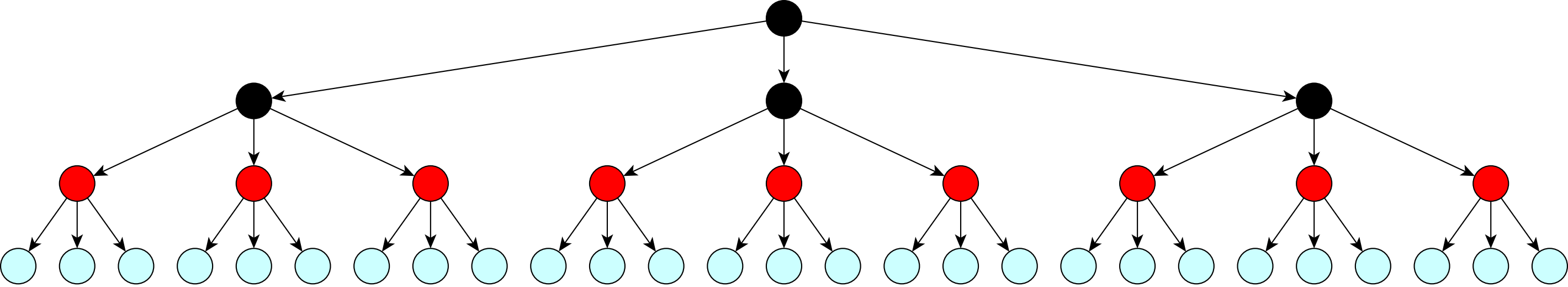


Algoritmo BFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

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



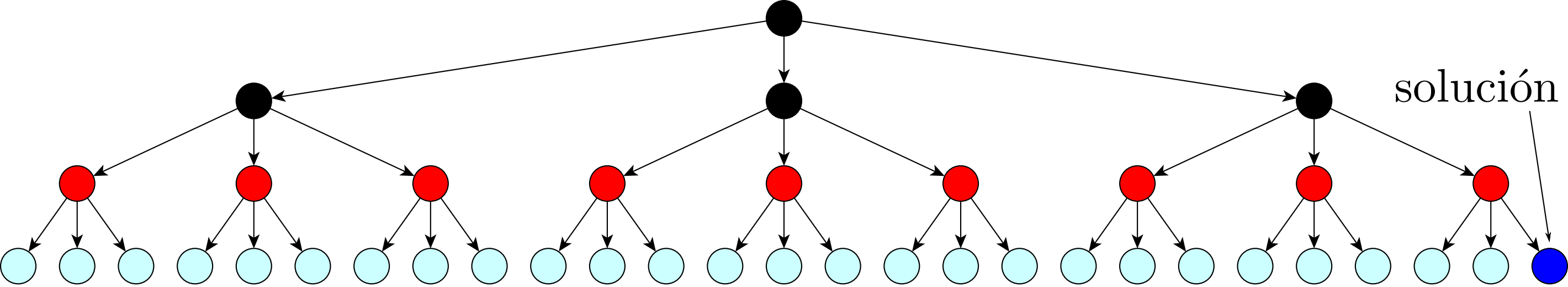
Algoritmo BFS

1. Memoria.

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0
1	3	1
2	9	4

A diferencia de DFS, la profundidad
de los nodos que se van expandiendo
crece monotonamente.



Algoritmo BFS

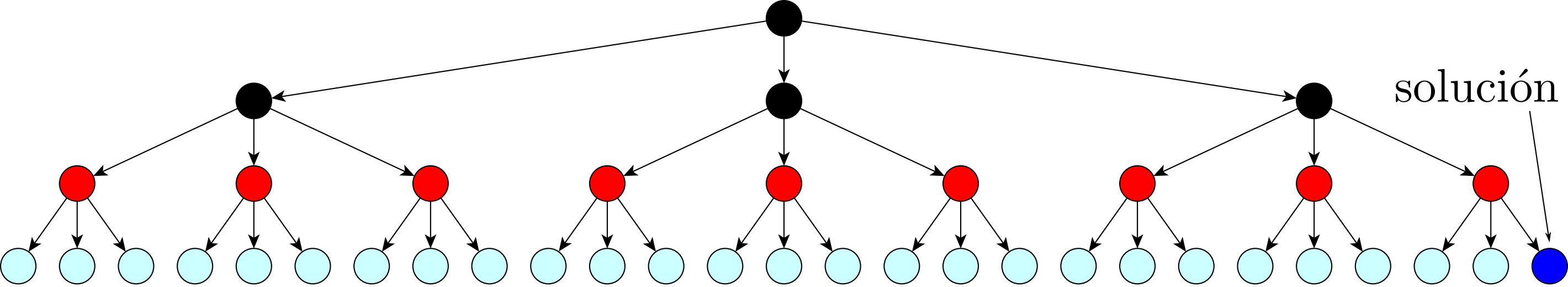
1. Memoria.

■ $O(b^d)$

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0
1	3	1
2	9	4
...		
n	b^n	$\frac{b^n - 1}{b - 1}$

El crecimiento de memoria para BFS
es exponencial con la profundidad,
pero acotada por la profundidad de
la solución.



Algoritmo BFS

1. Memoria.

- $O(b^d)$

2. Tiempo.

- $O(b^d)$

Observaremos como crece la frontera
y conjunto de expandidos

Profundidad (n)	Agenda	Expandidos
0	1	0
1	3	1
2	9	4
...		
n	b^n	$\frac{b^n - 1}{b - 1}$

El tiempo y la memoria crecen a la par.

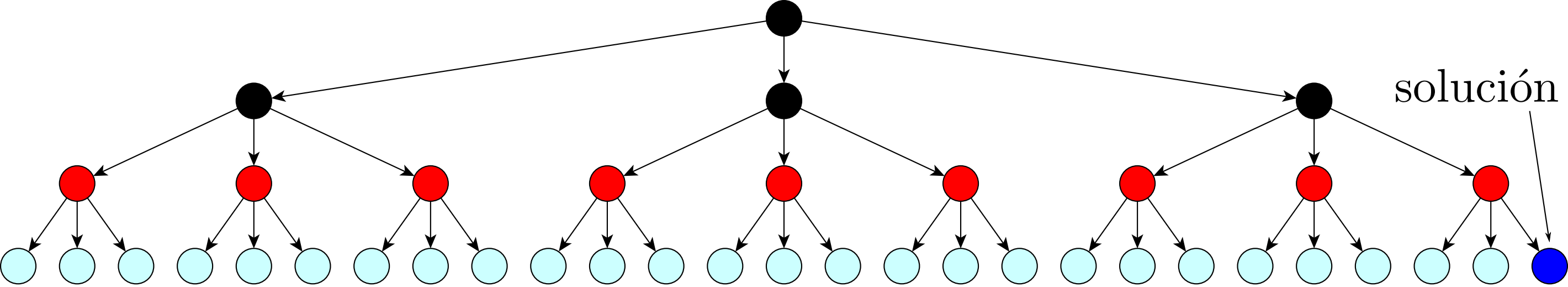


- $O(b^d)$

- $O(b^d)$

- Solución óptima.

Al agotar los niveles de manera incremental BFS encontrará la solución óptima en número de pasos.



Algoritmo BFS

1. Memoria.

- $O(b^d)$

2. Tiempo.

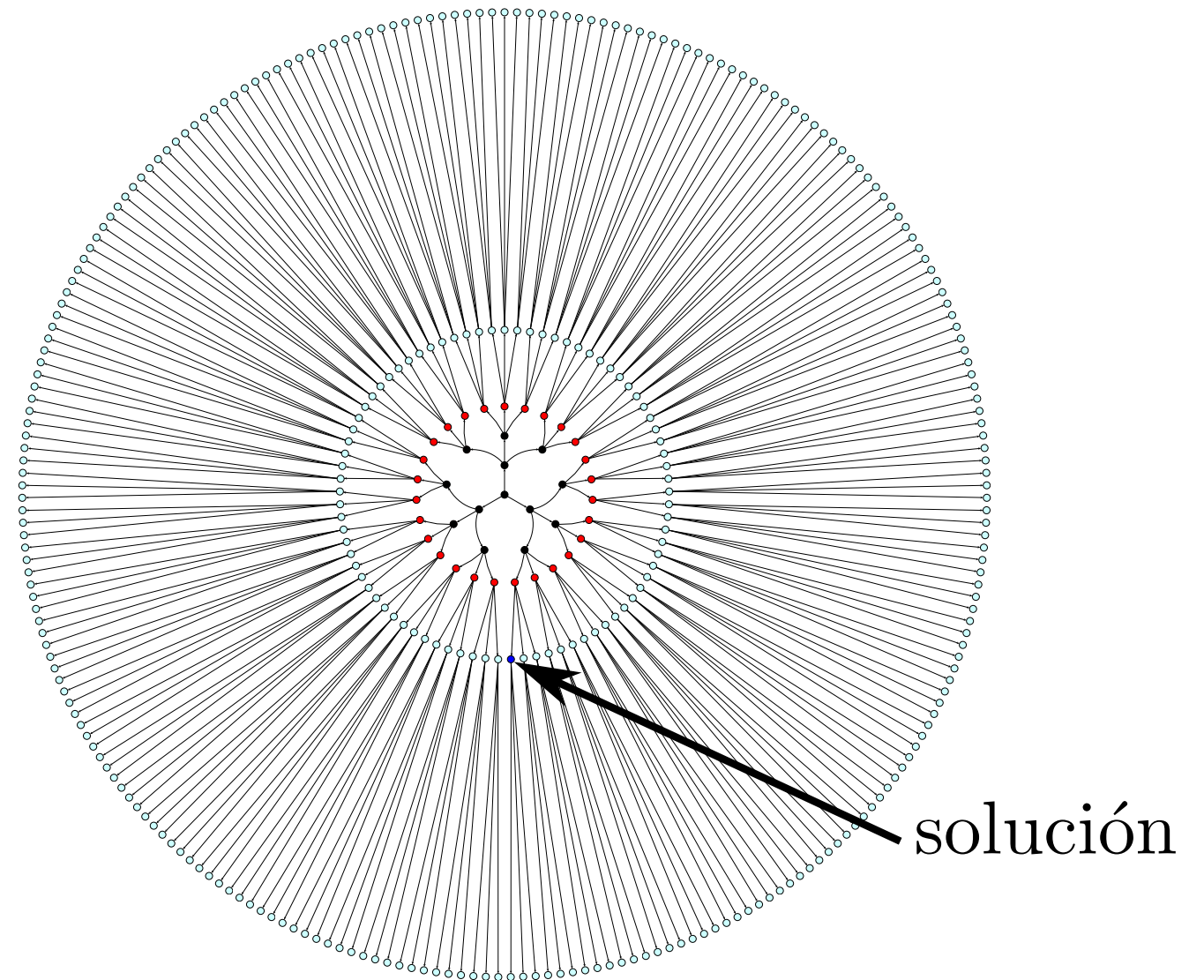
- $O(b^d)$

3. Calidad.

- Solución óptima.

4. Completez.

- Completo.



BFS encontrará la solución, sujeto a que disponga de los recursos computacionales suficientes.