

Inteligencia Artificial

Andrés Felipe Cruz Eraso

Institución Universitaria Colegio Mayor

Fuentes: Curso Inteligencia Artificial Universidad de Harvard
Inteligencia artificial por Stuart Russell y Peter Norvig

Inteligencia Artificial

Fundamentos filosóficos

Conceptos previos



Ingresa al siguiente enlace y realiza la encuesta:

<https://forms.gle/PfDEUFgJsSqumJwFA>

AGENDA 22/08/2024

- 1 OBJETIVOS
- 2 METODOLOGÍA DE EVALUACIÓN
- 3 CONCERTACIÓN DE CONTENIDOS
- 4 FUNDAMENTOS FILOSÓFICOS
- 5 DEFINICIÓN DE INTELIGENCIA ARTIFICIAL
- 6 EJEMPLOS DE USO
- 7 CARÁCTERISTICAS
- 8 LA PRUEBA DE TURING
- 9 **BUSQUEDA, CONOCIMIENTO, INCERTIDUMBRE, OPTIMIZACIÓN**

OBJETIVOS

General: Introducir a los orígenes de la inteligencia artificial, aplicaciones y algoritmos subyacentes

Específicos:

Revisar aspectos filosóficos acerca del tema de inteligencia artificial

Propuesta de contenidos para temática de inteligencia artificial

Listar algoritmos más usados en IA y explicar variaciones de algoritmo de búsqueda y realizar práctica con tema mas sugerido.

METODOLOGÍA DE EVALUACIÓN

Específicos:

Exposiciones 25 %

Exámen en línea 40 %

Prácticas 25 %

Asistencia 5

Participación %

PROPUESTA CONTENIDOS

Capítulo 1

Origen e historia

Conceptos básicos de la Inteligencia Artificial (IA)

Áreas de estudio de la Inteligencia Artificial

PROPUESTA CONTENIDOS

Capítulo 2. Solución a problemas de búsqueda
Solución a problemas mediante espacio de
estados
Estrategias de búsqueda informada
Estrategias de búsqueda no informada

PROPUESTA CONTENIDOS

Capítulo 3: Problemas de optimización
Introducción a los problemas de optimización
Definición de algoritmos genéticos
Operadores para representación binaria de algoritmos genéticos
Operadores para representación permutada de algoritmos genéticos

PROPUESTA CONTENIDOS

Capítulo 4: Machine Learning

Introduction a Machine Learning

Metodología Crisp ML

Problemas de clasificación: Redes Neuronales

Problemas de agrupamiento: Clustering

PROPUESTA CONTENIDOS

Capítulo 5: Sistemas Basados en Reglas
Sistemas de Logica difusa
Sistemas Expertos

PROPUESTA CONTENIDOS ADICIONALES

Capítulo 2-0 Conocimiento

Lógica proposicional

Base de conocimiento

Inferencia, algoritmos de inferencia, modelos de chequeo, ingeniería de conocimiento, reglas de inferencia, resolución, lógica de primer orden, cuantificación universal

PROPUESTA CONTENIDOS ADICIONALES

Capítulo 2 Conocimiento

Lógica proposicional

Base de conocimiento

Inferencia, algoritmos de inferencia, modelos de chequeo, ingeniería de conocimiento, reglas de inferencia, resolución, lógica de primer orden, cuantificación universal

PROPUESTA CONTENIDOS ADICIONALES

Capítulo n Incertidumbre

Probabilidad

Regla de bayes

Probabilidad conjunta

Reglas de probabilidad, inferencia, redes bayesianas,
inferencia, muestreo, rechazo de muestreo, sensores,
cadena de markov, modelos de sensores, incertidumbre

Inteligencia Artificial

Fundamentos filosóficos



Inteligencia Artificial

Fundamentos filosóficos



Inteligencia Artificial

Fundamentos filosóficos



Inteligencia Artificial

Fundamentos filosóficos



Inteligencia Artificial

Fundamentos filosóficos



Inteligencia Artificial

Fundamentos filosóficos



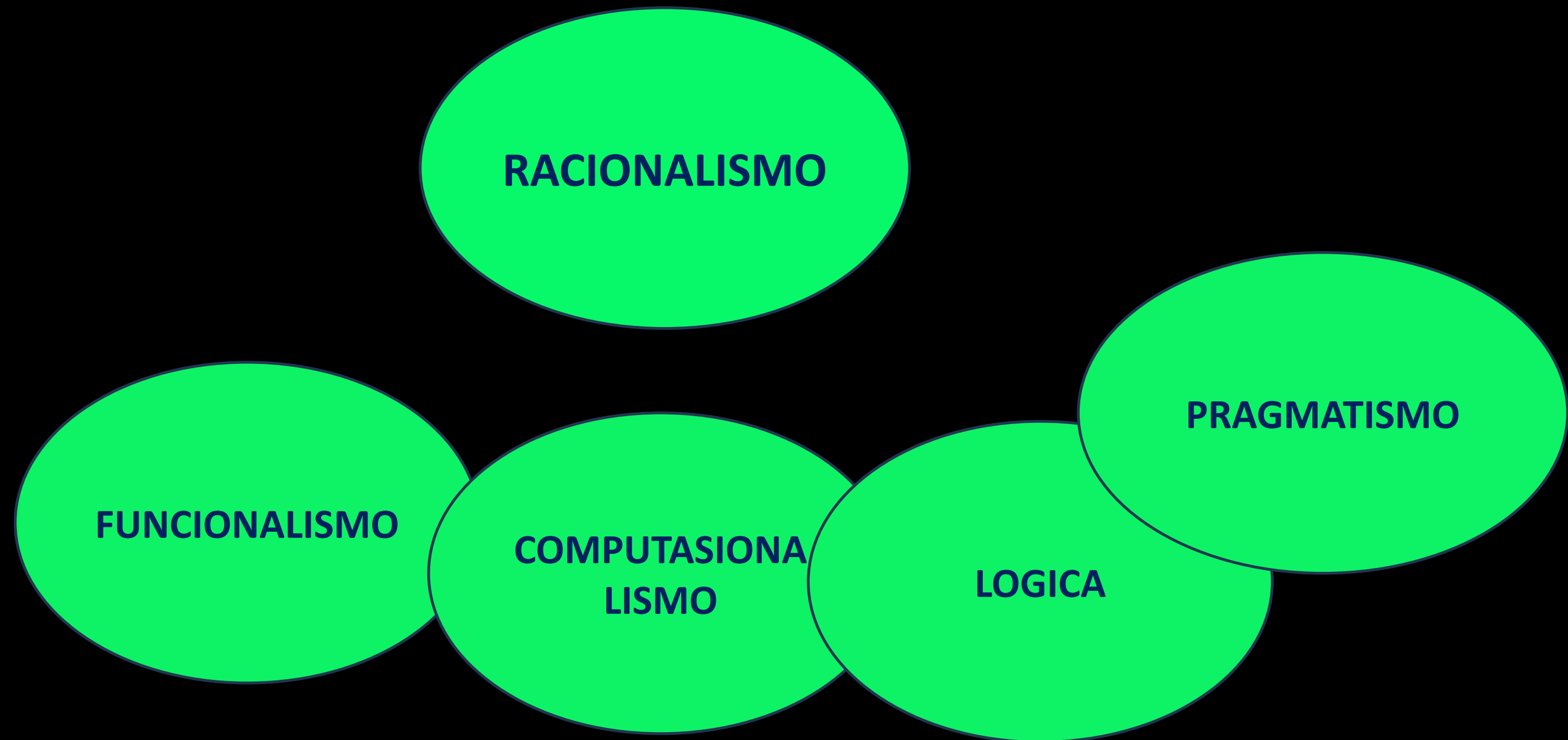
Inteligencia Artificial

Fundamentos filosóficos



Inteligencia Artificial

Fundamentos filosóficos



Inteligencia Artificial

Definición

La inteligencia artificial (IA) es una rama de la informática que se centra en la creación de sistemas capaces de realizar tareas que, cuando son realizadas por seres humanos, requieren de inteligencia. Estas tareas pueden incluir el reconocimiento de patrones, la toma de decisiones, la resolución de problemas, el aprendizaje, la comprensión del lenguaje natural, y la percepción visual, entre otras.

Partidas de ajedrez de Deep Blue contra gasparov

Inteligencia Artificial

Aplicaciones - Ejemplos

Detección de enfermedades, diabetes

Carros Autónomos

Recomendación de contenido

Procesamiento de imagen y video

Robótica en Google 0 en Recolección de coliflor

Inteligencia Artificial

CARACTERÍSTICAS

Autonomía

La capacidad para ejecutar tareas en situaciones complejas sin la dirección constante del usuario.

Adaptabilidad

La capacidad para mejorar la ejecución de las tareas aprendiendo de la experiencia.

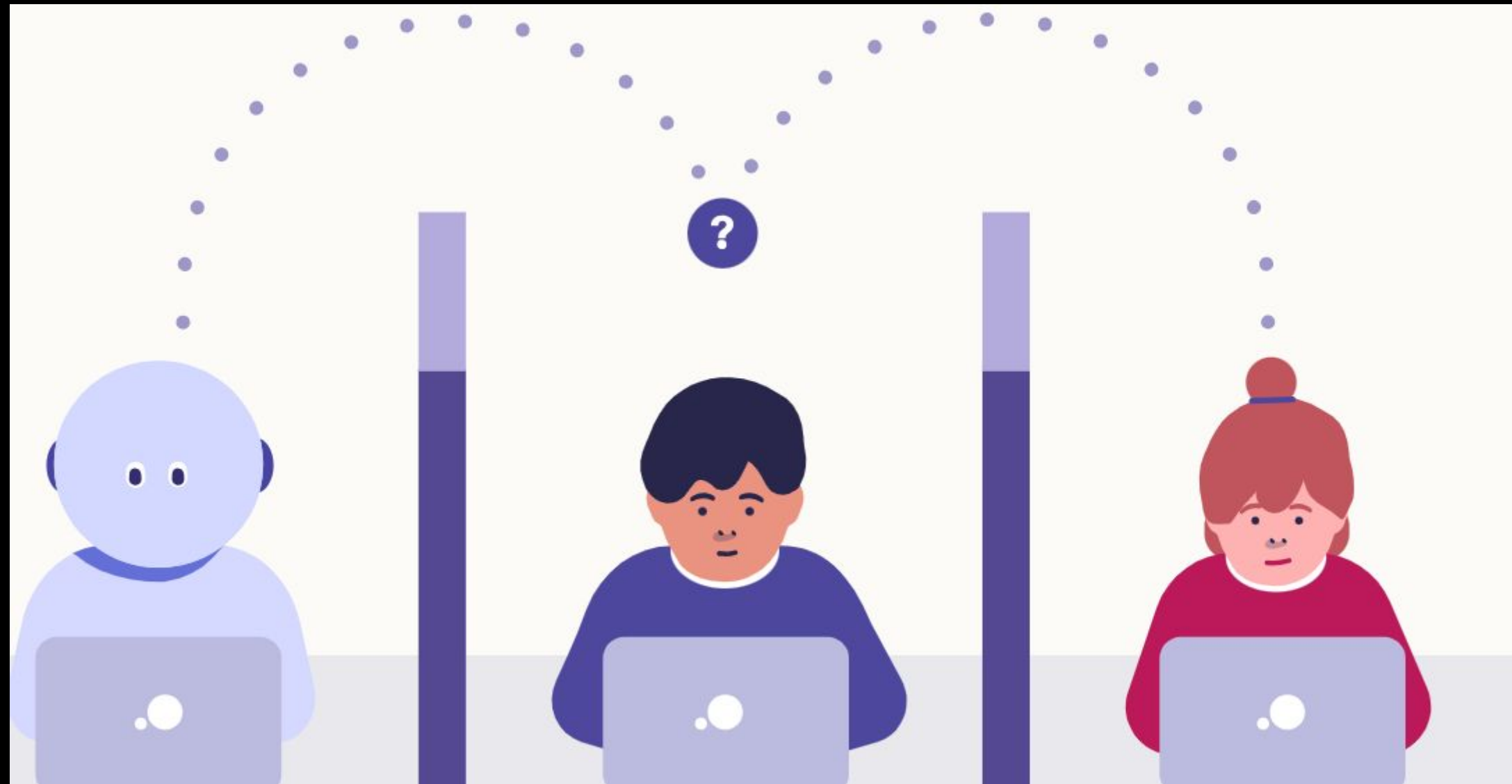
Inteligencia Artificial

La prueba de Turing

Alan Turing (1912-1954) fue un matemático y lógico inglés. Se le considera, con razón, el padre de la informática. A Turing le fascinaban la inteligencia y el pensamiento, y la posibilidad de simularlos mediante máquinas. La contribución más destacada de Turing a la IA es su juego de imitación, que más tarde se conoció como el test de Turing.

Inteligencia Artificial

La prueba de Turing



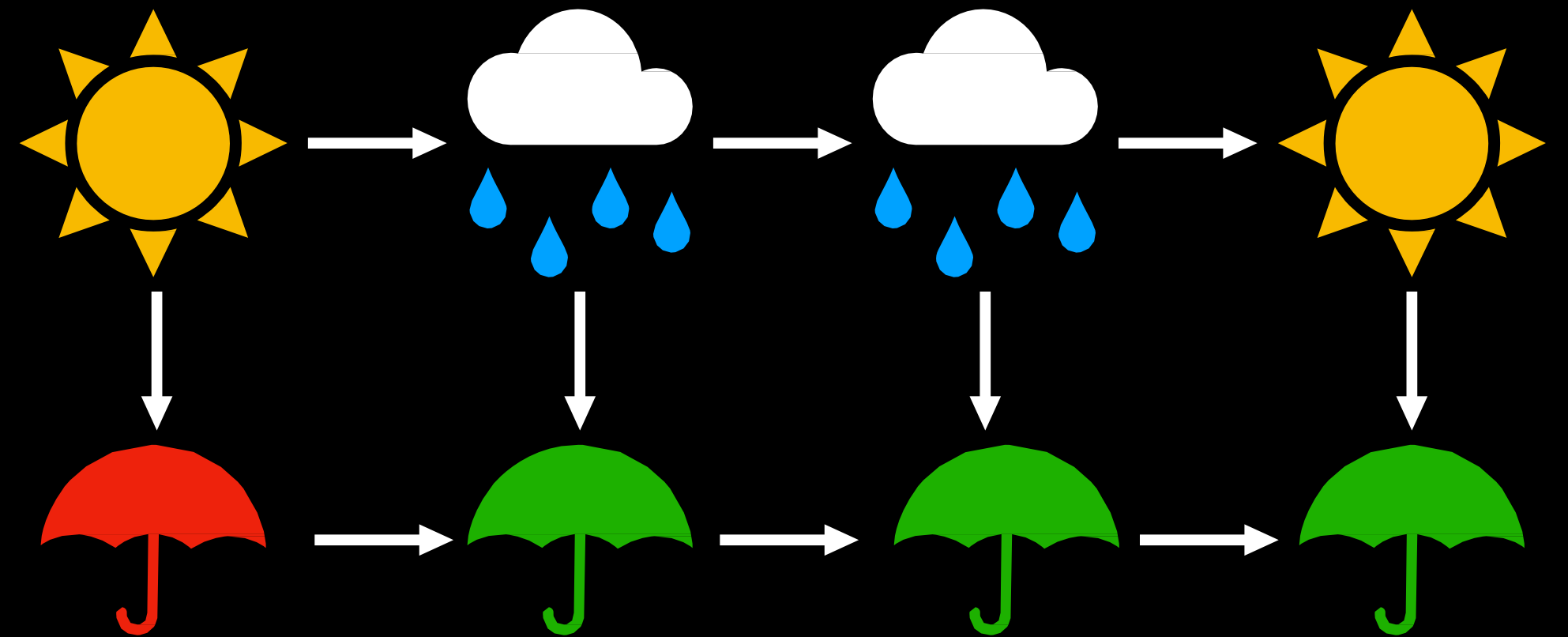
Búsqueda

O		
	X	X
	O	X

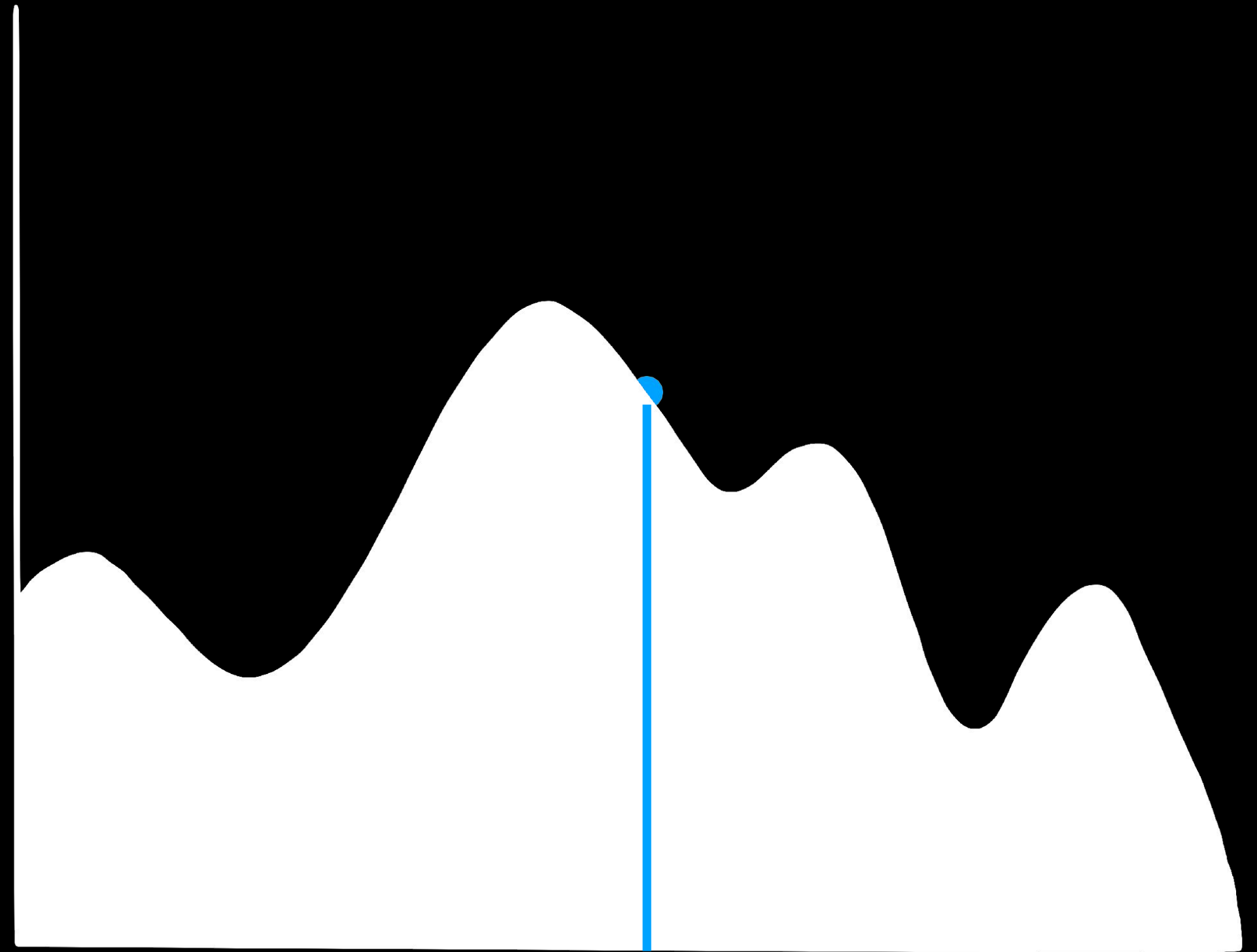
Conocimiento

$$\frac{P \rightarrow Q}{Q}$$

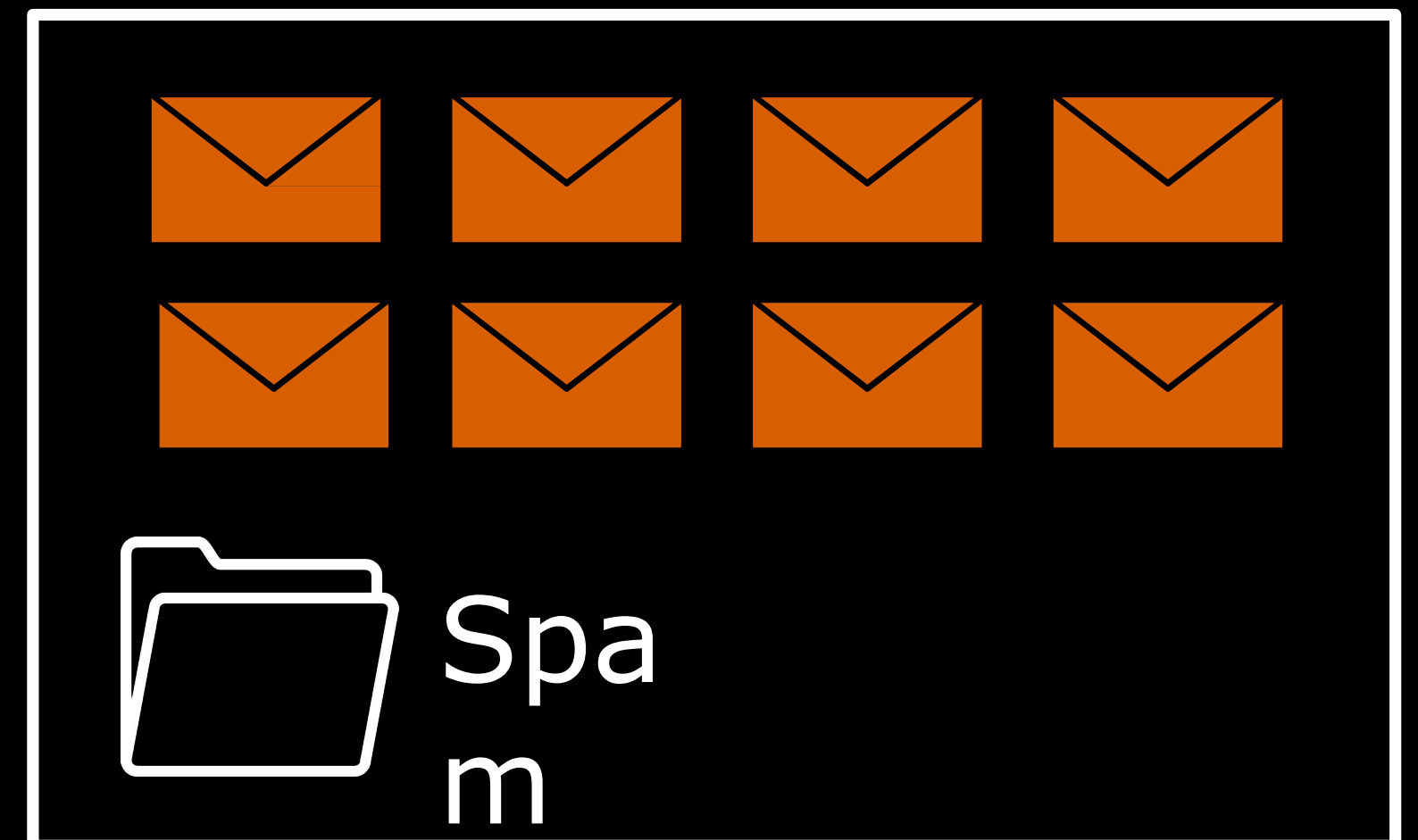
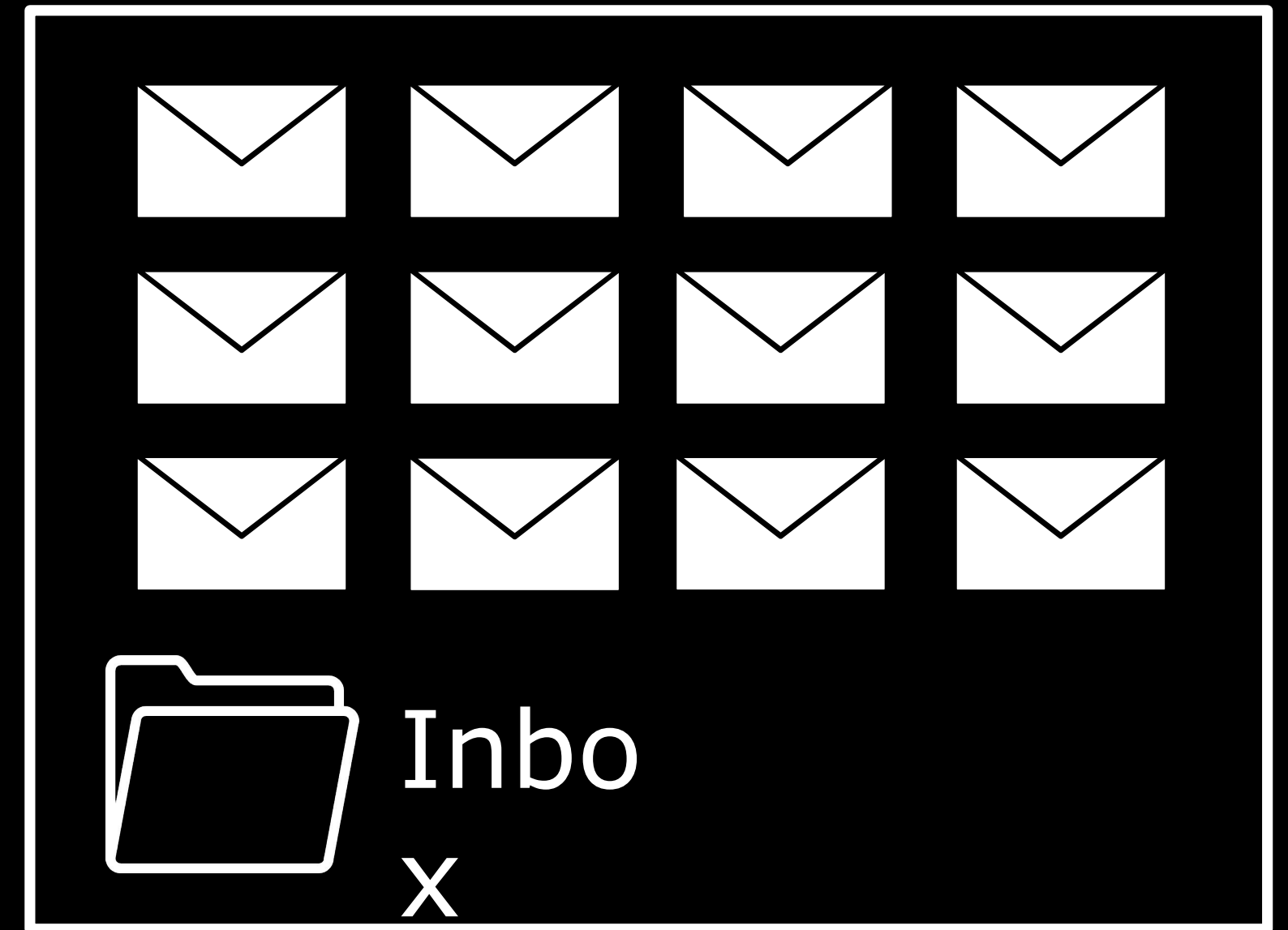
Incertidumbre



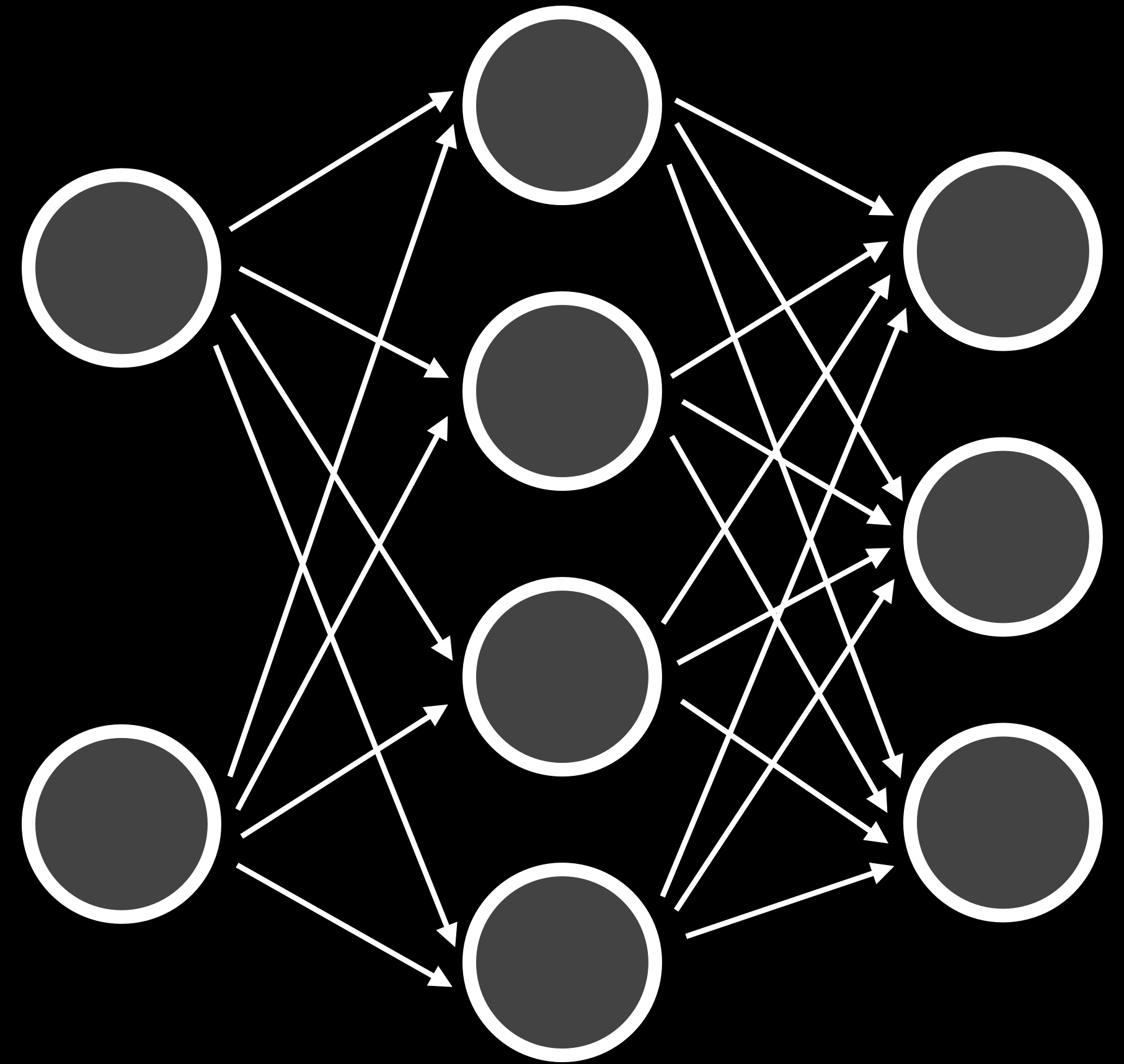
Optimización



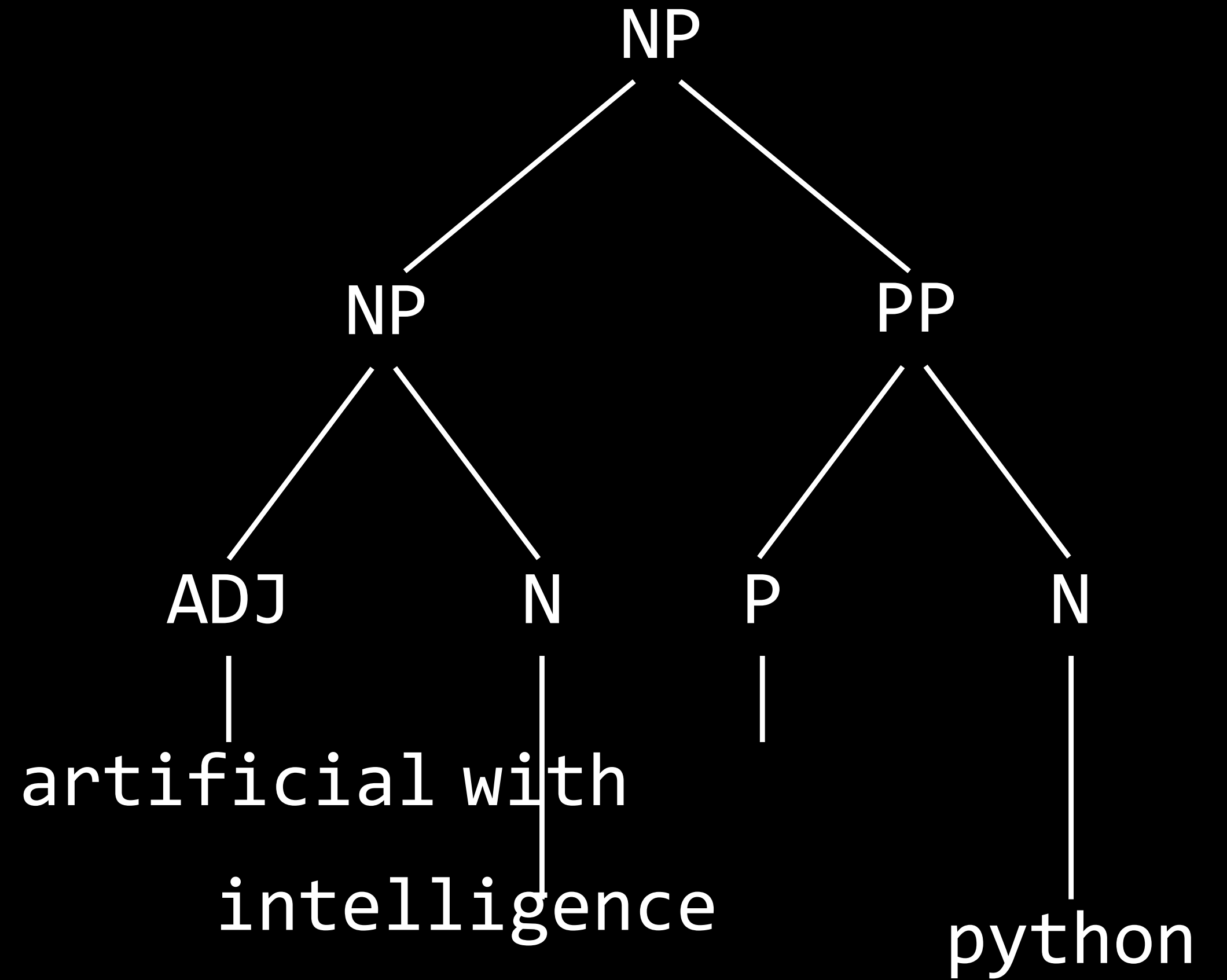
Conocimiento



Redes Neuronales



Lenguaje



Búsqueda

1

2

3

4

5

6

7

8

9

10

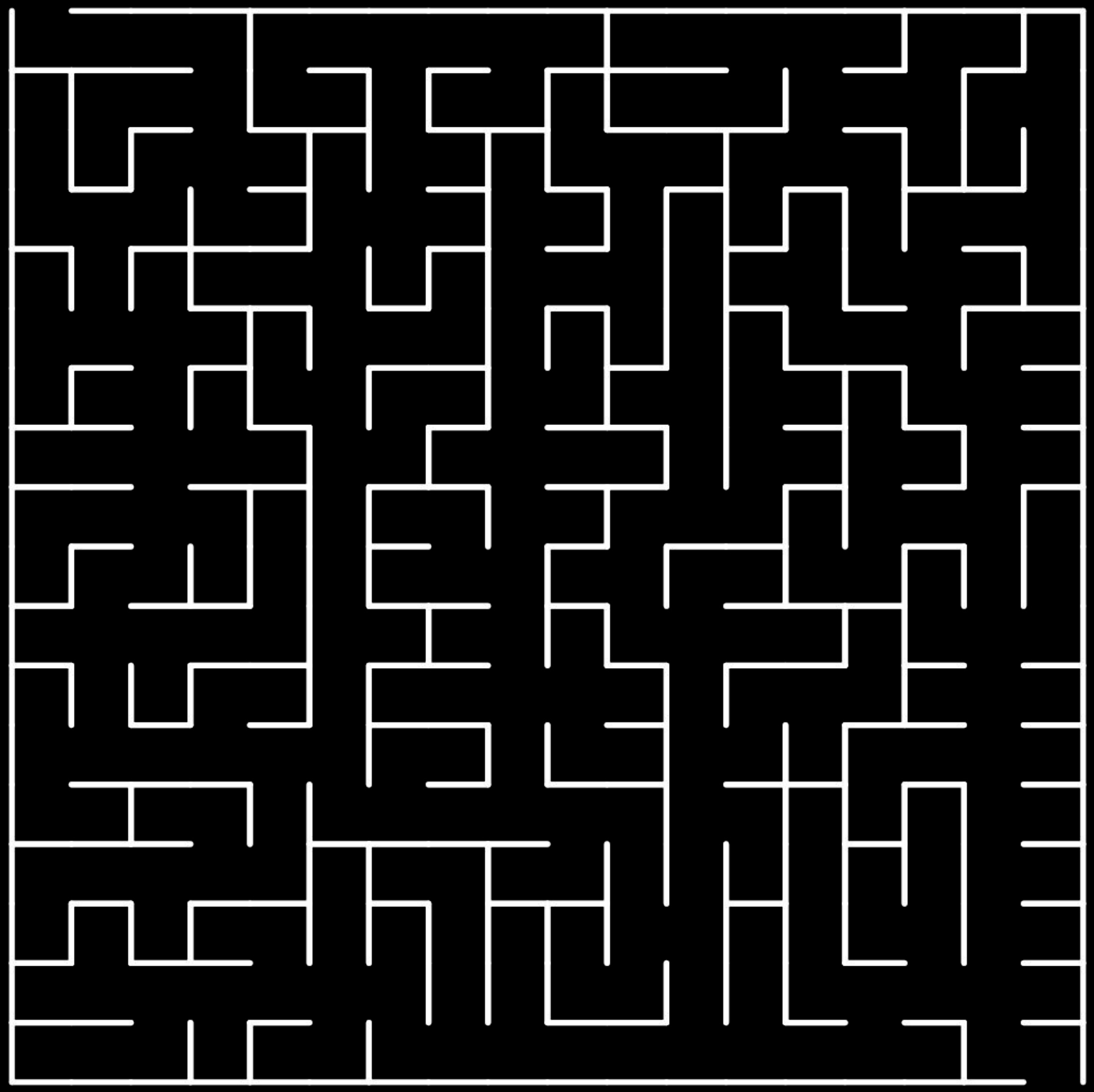
11

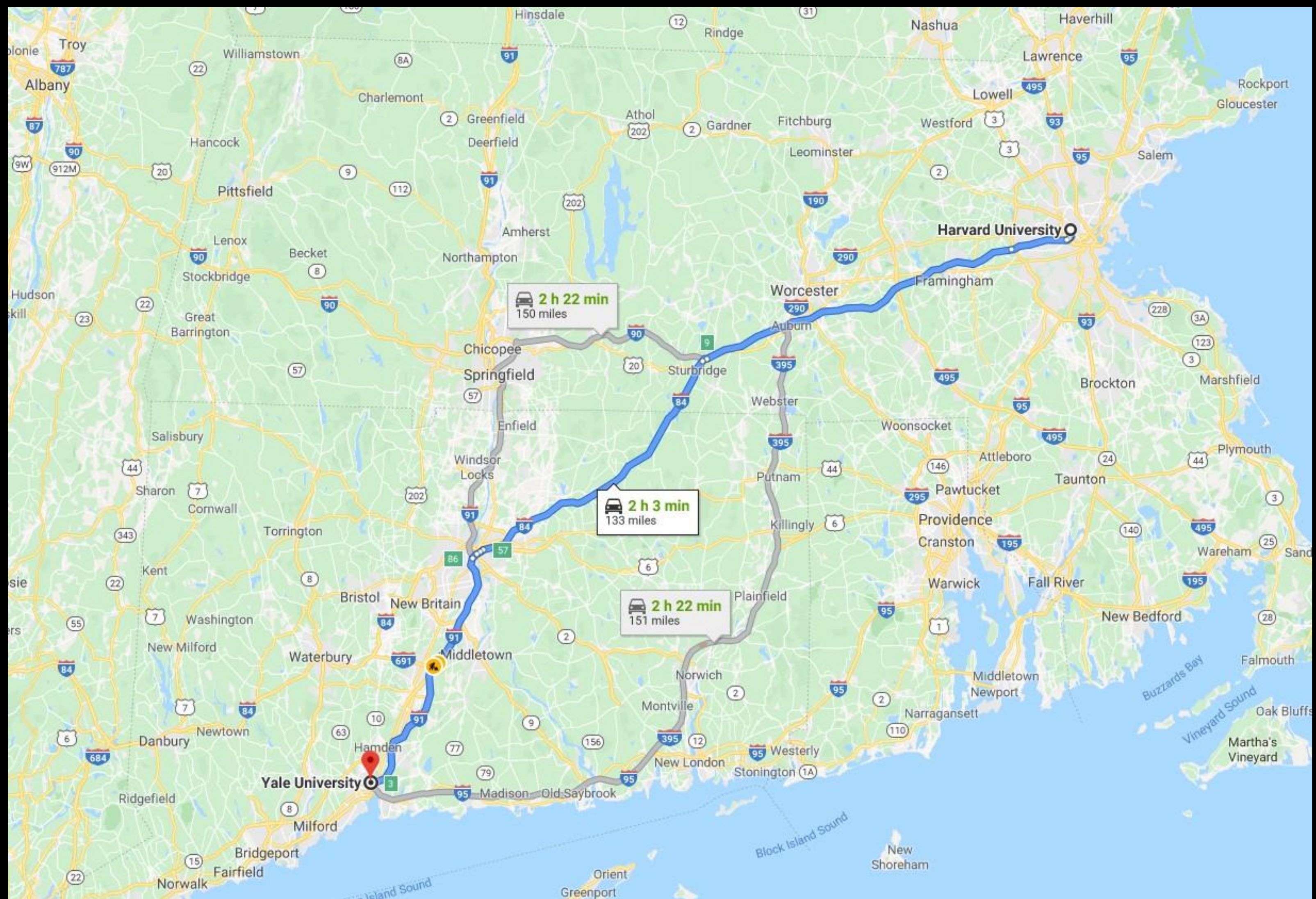
12

13

14

15





Problemas de búsqueda

Agente

entidad que percibe su entorno y actúa
sobre él

Estado

La configuración en un momento del tiempo del agente y su entorno

2	4	5	7
8	3	1	11
14	6		10
9	13	15	12



Estado inicial

El estado en el cual inicia el agente

Estado inicial

2	4	5	7
8	3	1	11
14	6		10
9	13	15	12

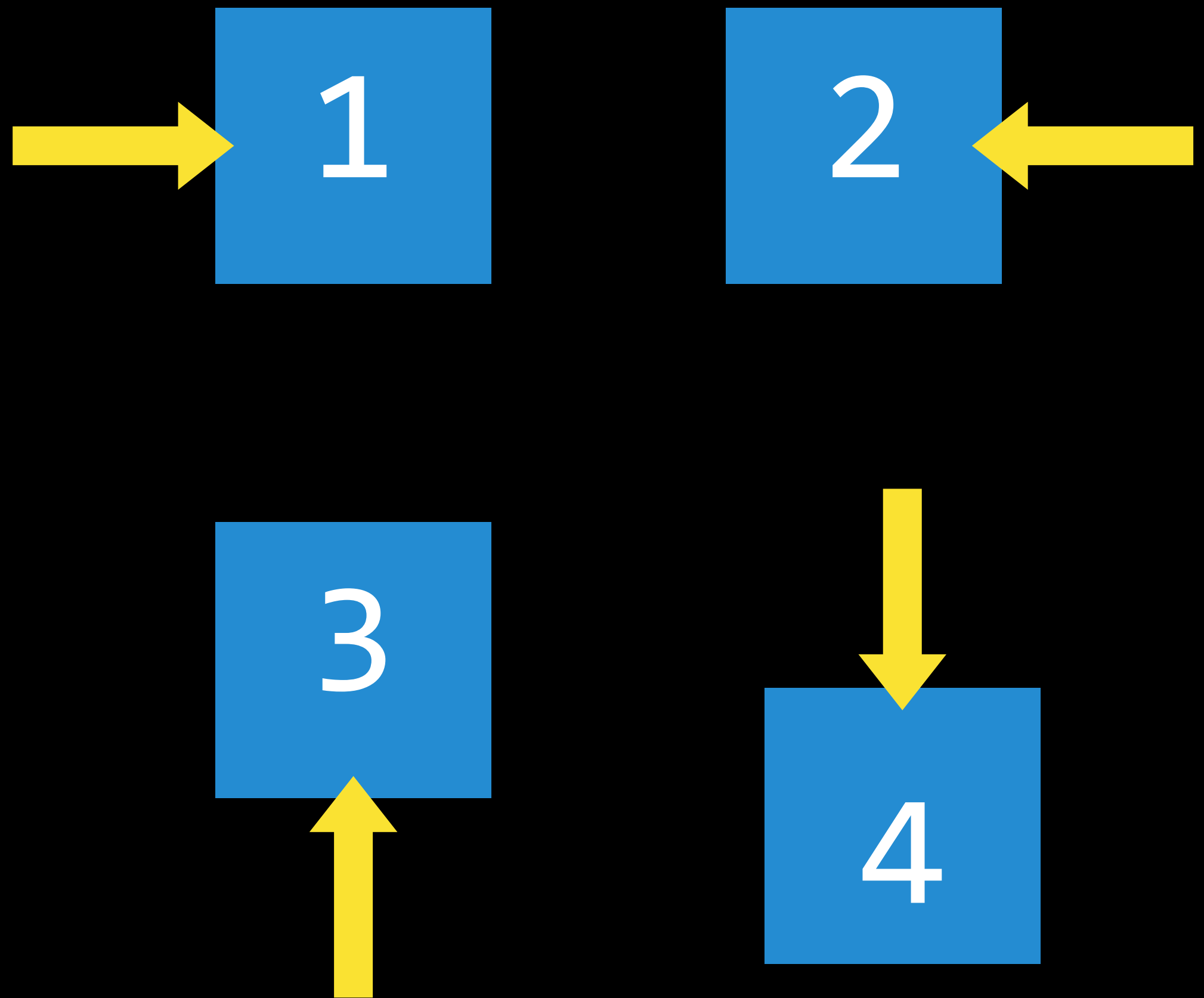
Acciones

Elecciones que pueden ser hechas en
un estado

Acciones

$\text{ACCIONES}(s)$ retorna el conjunto de acciones que pueden ser ejecutadas en un estado s

acciones



Modelo de transición

Una descripción de que estado resulta de ejecutar cualquier aplicación aplicable en cualquier estado

Modelo de transición

$\text{resultado}(s, a)$ Retorna el estado resultante de ejecutar una acción a en un estado s

Resultado
(


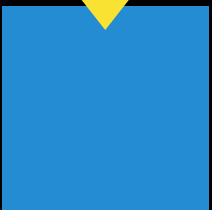
2	4	5	7
8 3	1 11		
14	6	10	12
9	13	15	

,  ) =

2	4	5	7
8 3	1 11		
14	6	10	12
9	13		15

Resultado(

2	4	5	7
8 3	1 11		
14	6	10	12
9	13	15	

,  ) =

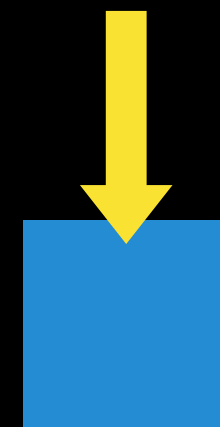
8 3	1 11		
9	13	15	12

Modelo de transición

Resultado(

2	4	5	7
8 3	1 11		
14	6	10	12
9	13	15	

,

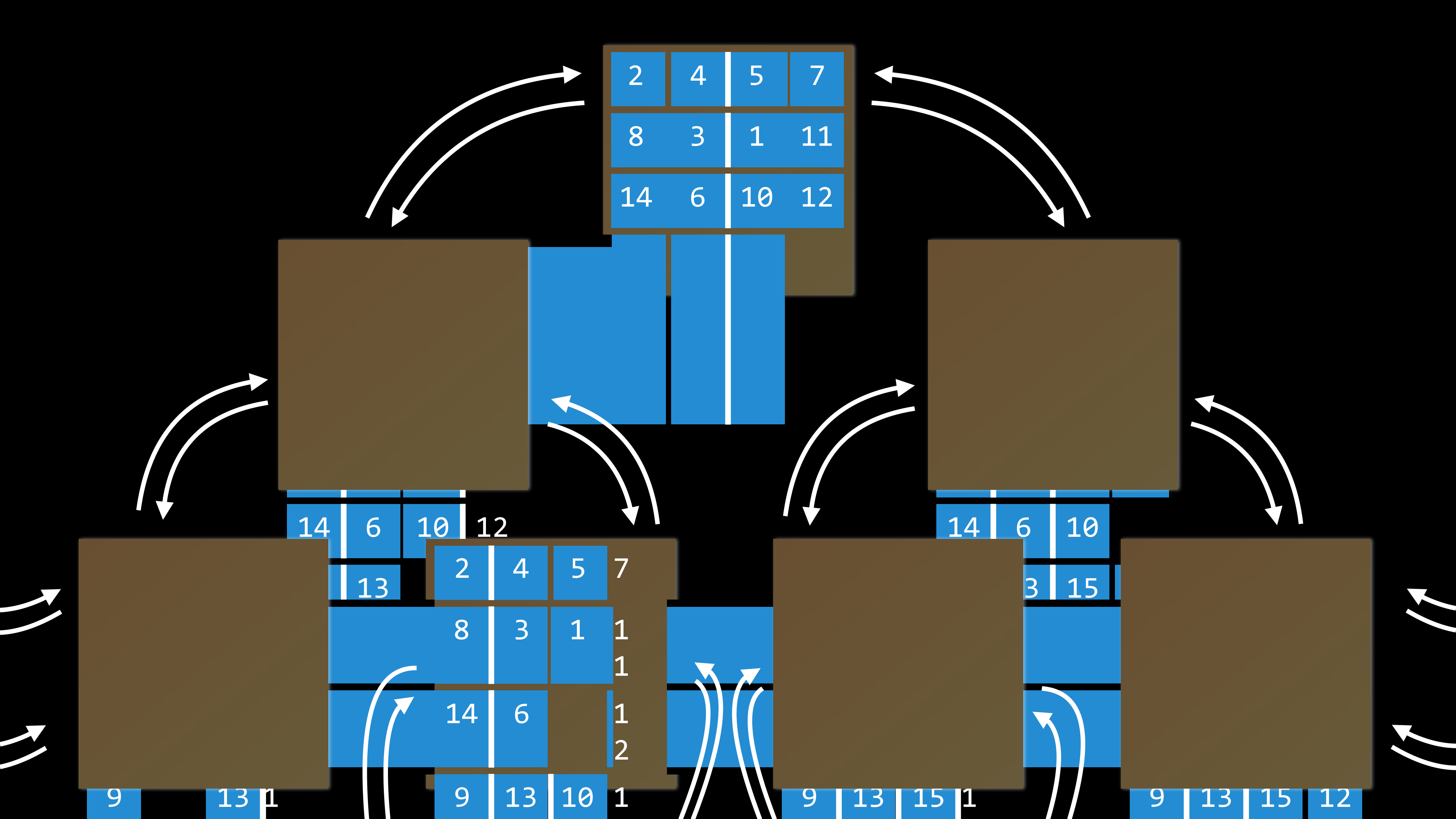


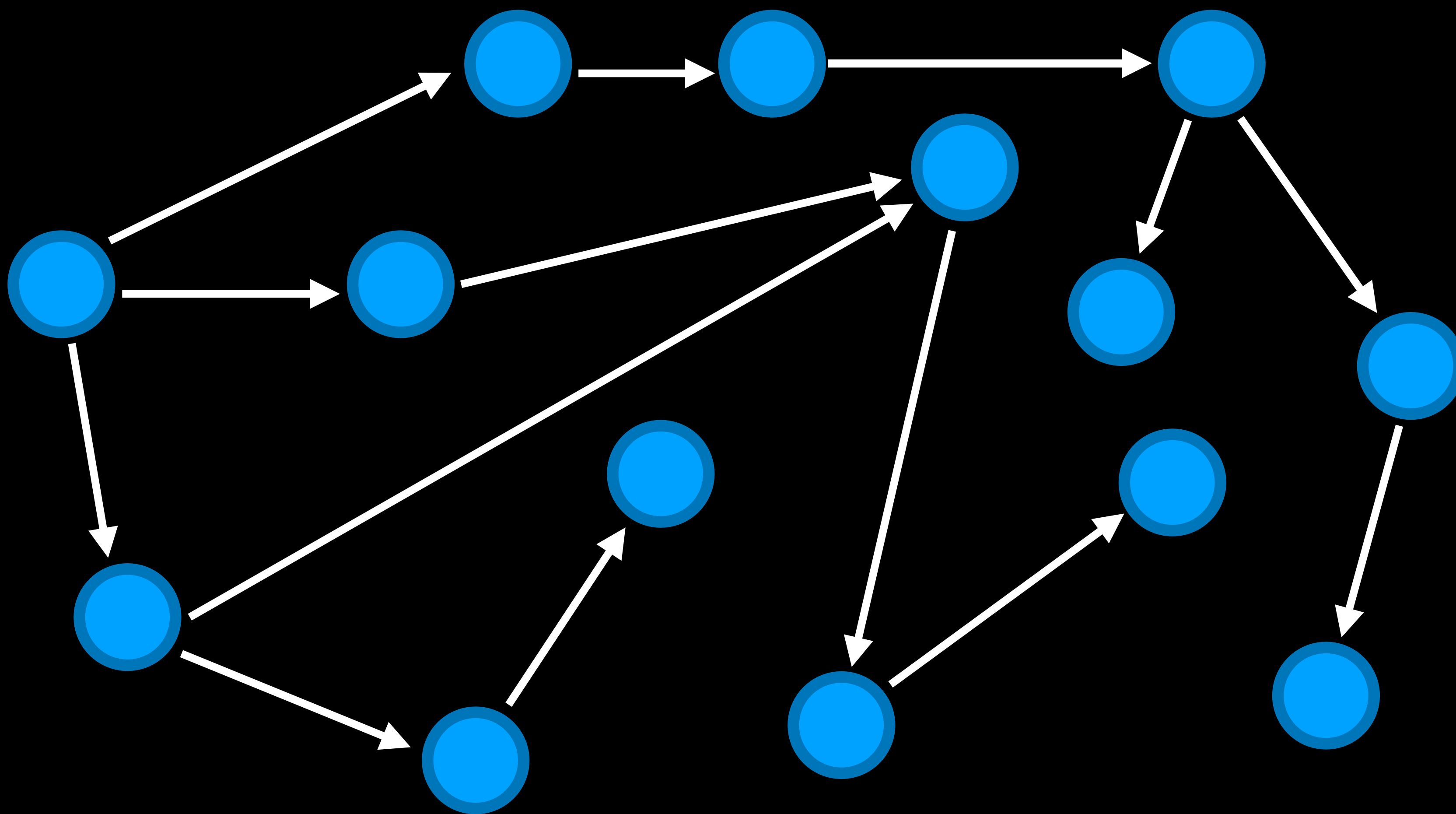
) =

2	4	5	7
8 3	1 11		
9	13	15	12

Espacio de estados

Es el conjunto de estados alcanzables desde el estado inicial por una secuencia de acciones



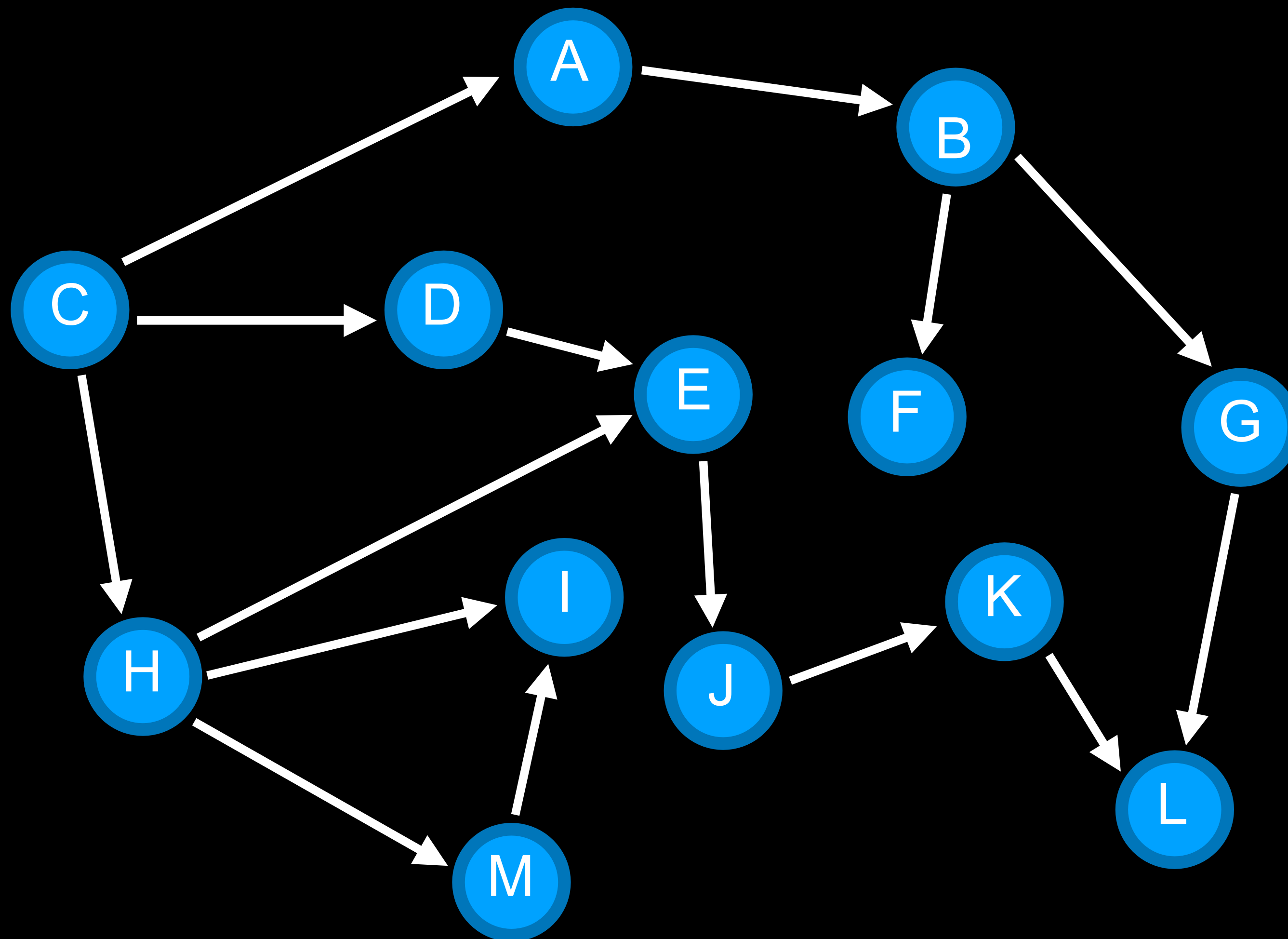


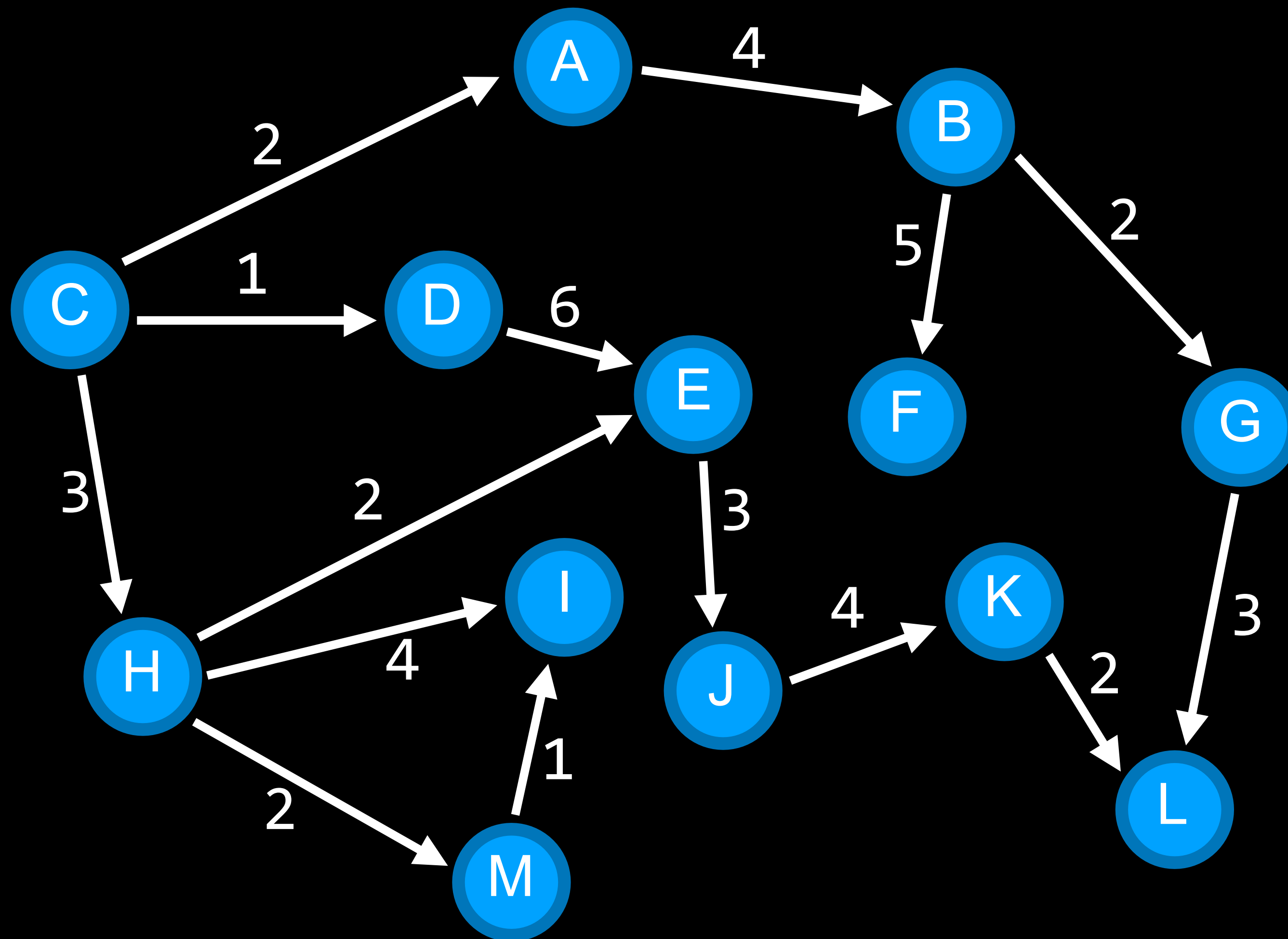
Prueba de estado

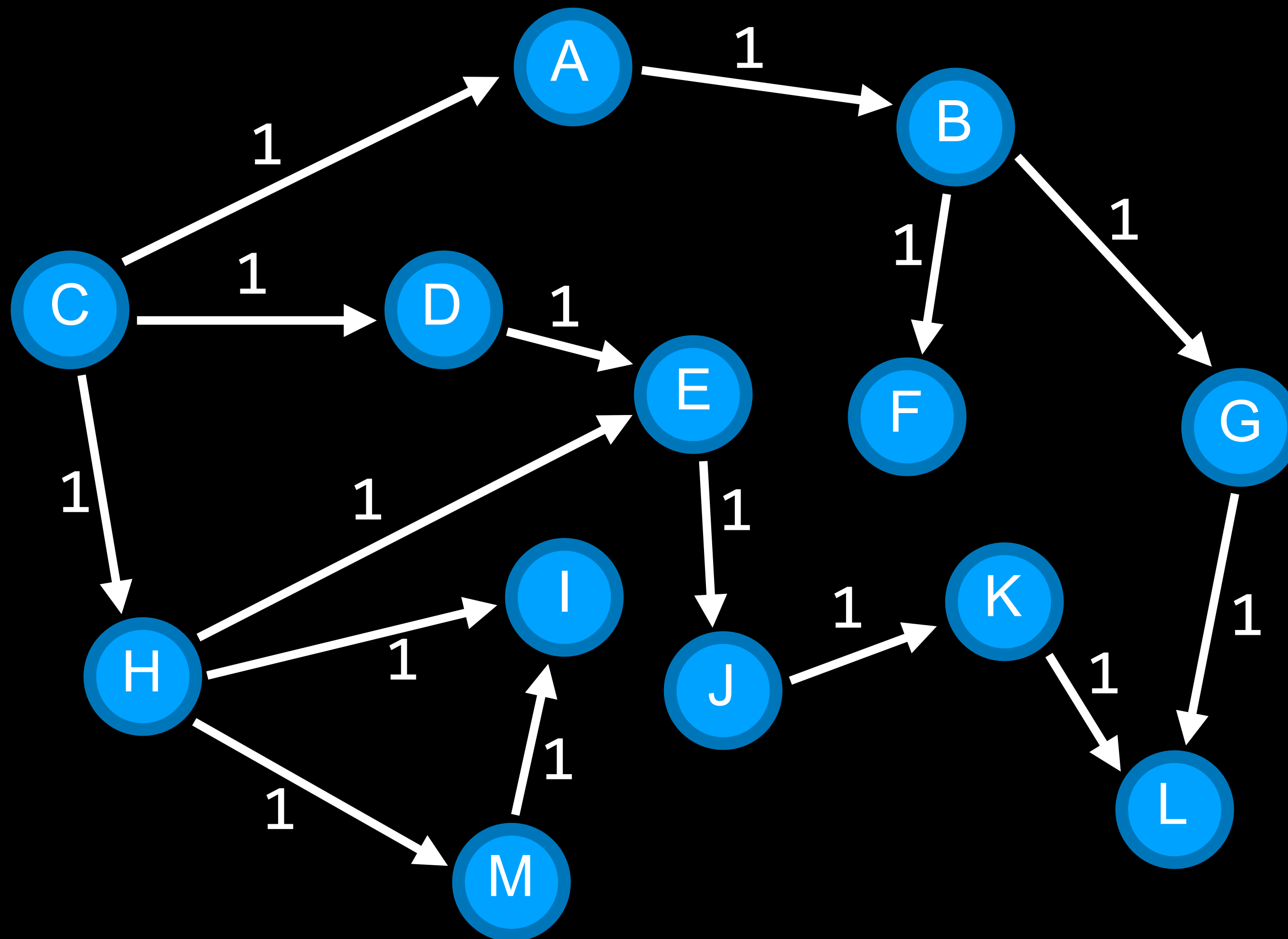
Es la manera de establecer si un estado dado es el estado meta

Costo de ruta

Costo numérico asociado con una ruta dada







Problemas de búsqueda

- Estado inicial
- acciones
- Modelo de transición
- Evaluación de meta
- Función de costo de ruta

Solución

Es un conjunto de acciones que nos llevan desde un estado inicial hasta un estado meta

Solución óptima

La solución con el menor costo de ruta
entre las posibles rutas de solución

nodo

Es una estructura de datos que mantiene un registro de:

- Un estado
- un **padre** (nodo que genera este)
- una **accion** (accion aplicada al padre para obtener un nodo)
- **costo de ruta** (del estado inicial estado hasta un nodo)

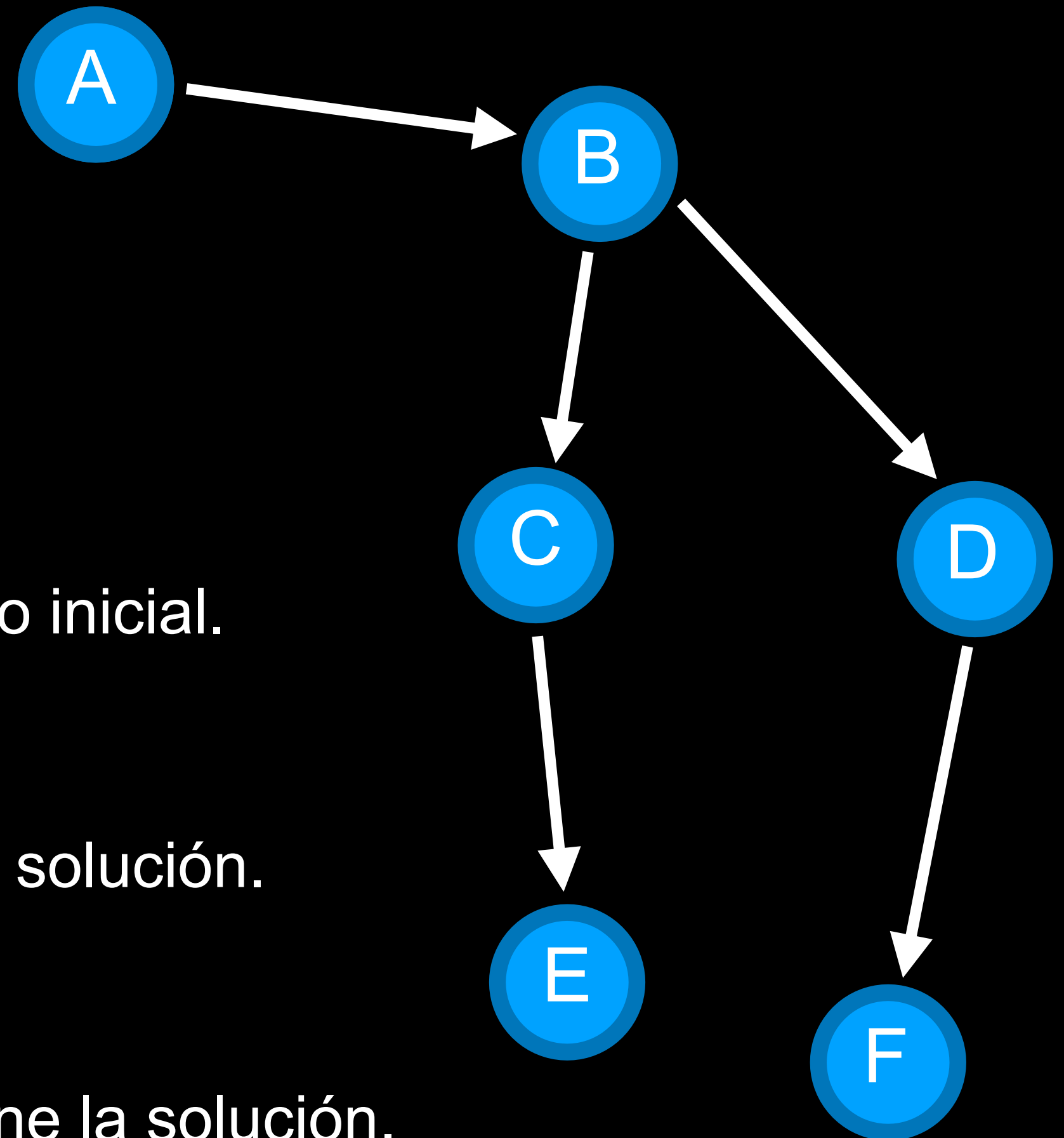
Aproximación

- Inicia con una **frontera** que contiene el estado inicial.
- Repetir:
 - Si la frontera está vacía entonces no hay solución.
 - Renovar un nodo de la frontera.
 - Si el nodo contiene el estado meta, retornar la solución.
 - *Adicionar el nodo al conjunto explorado*
- **Expandir** node, adicionar los nodos resultantes a la frontera, si no están allí o el conjunto explorado.

Camino de A hasta E?

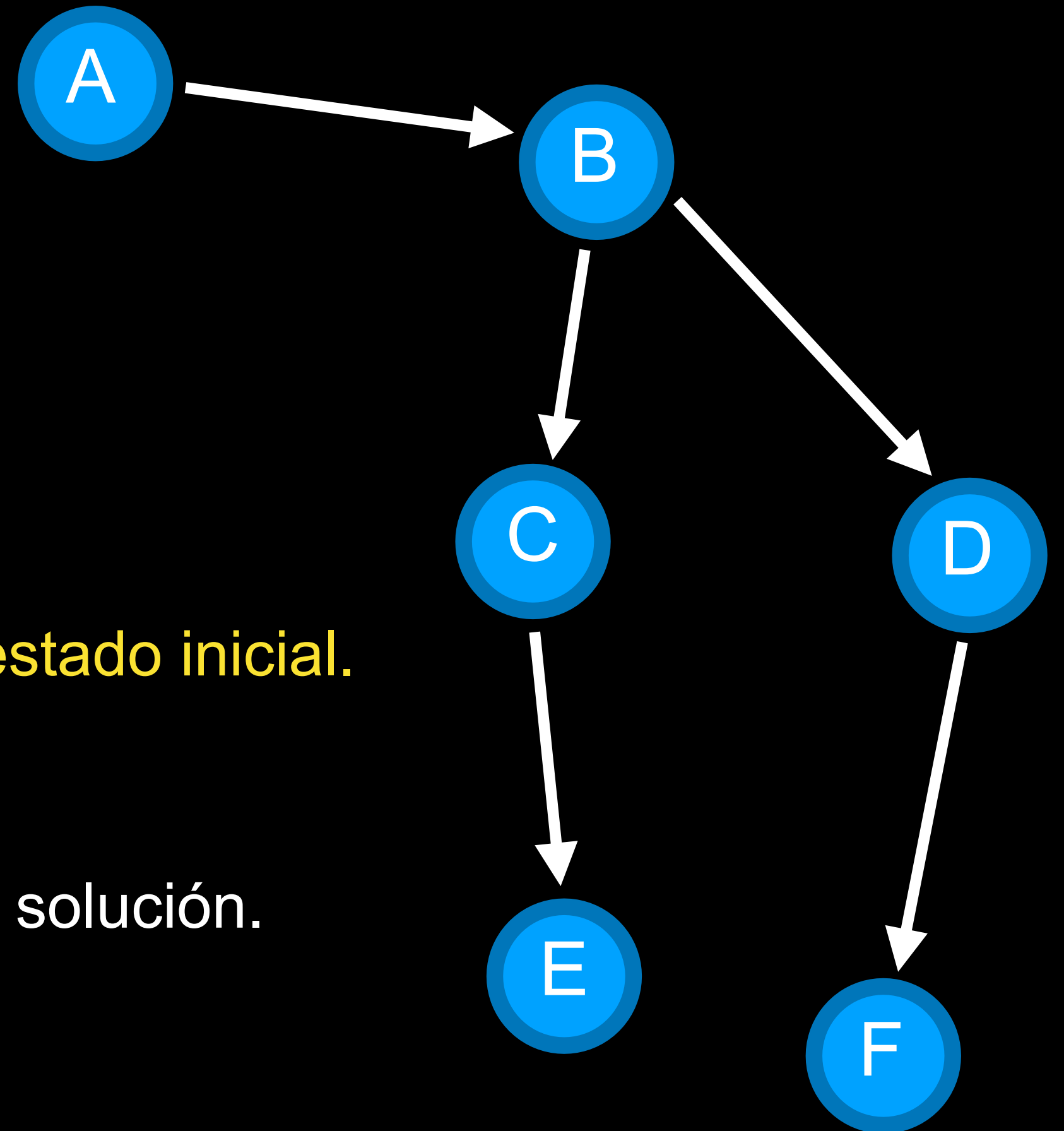
Frontera

- Comience con una frontera que tenga el estado inicial.
- Repetir:
 - Si la frontera está vacía, entonces no hay solución.
 - Remove un nodo desde la frontera.
 - Si el nodo contiene un estado meta, retorne la solución.
 - **Expandir** nodo, adicionar nodos resultantes a la frontera.



Camino de A a E?

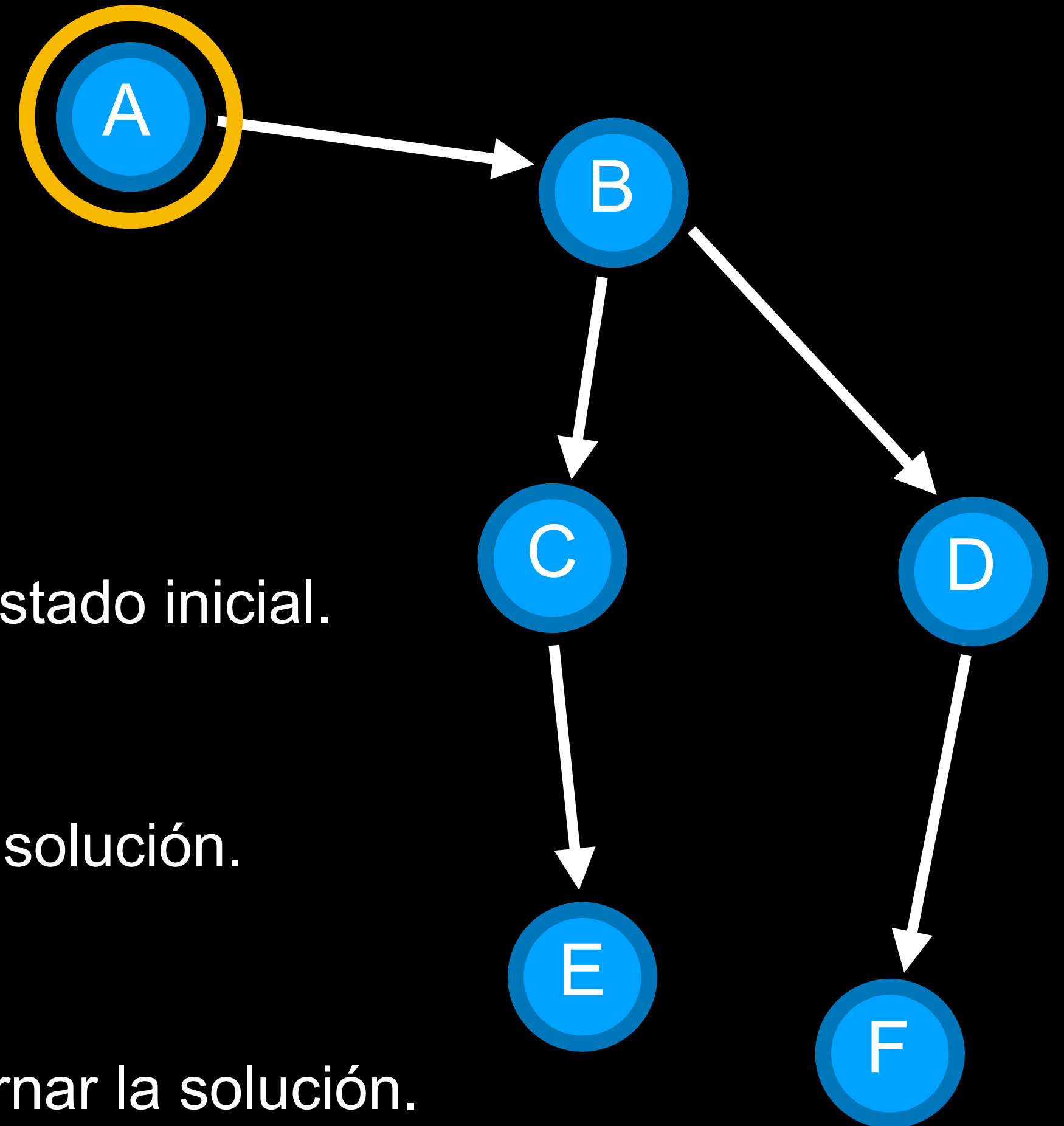
Frontera



- Comenzar con una **frontera** que contiene el estado inicial.
- Repetir:
 - Si la frontera está vacía, entonces no hay solución.
 - Remover un nodo de la frontera.
 - Si , es el nodo meta entonces retornar la solución.
 - **Expand** nodo, añadir nodos resultantes a la frontera.

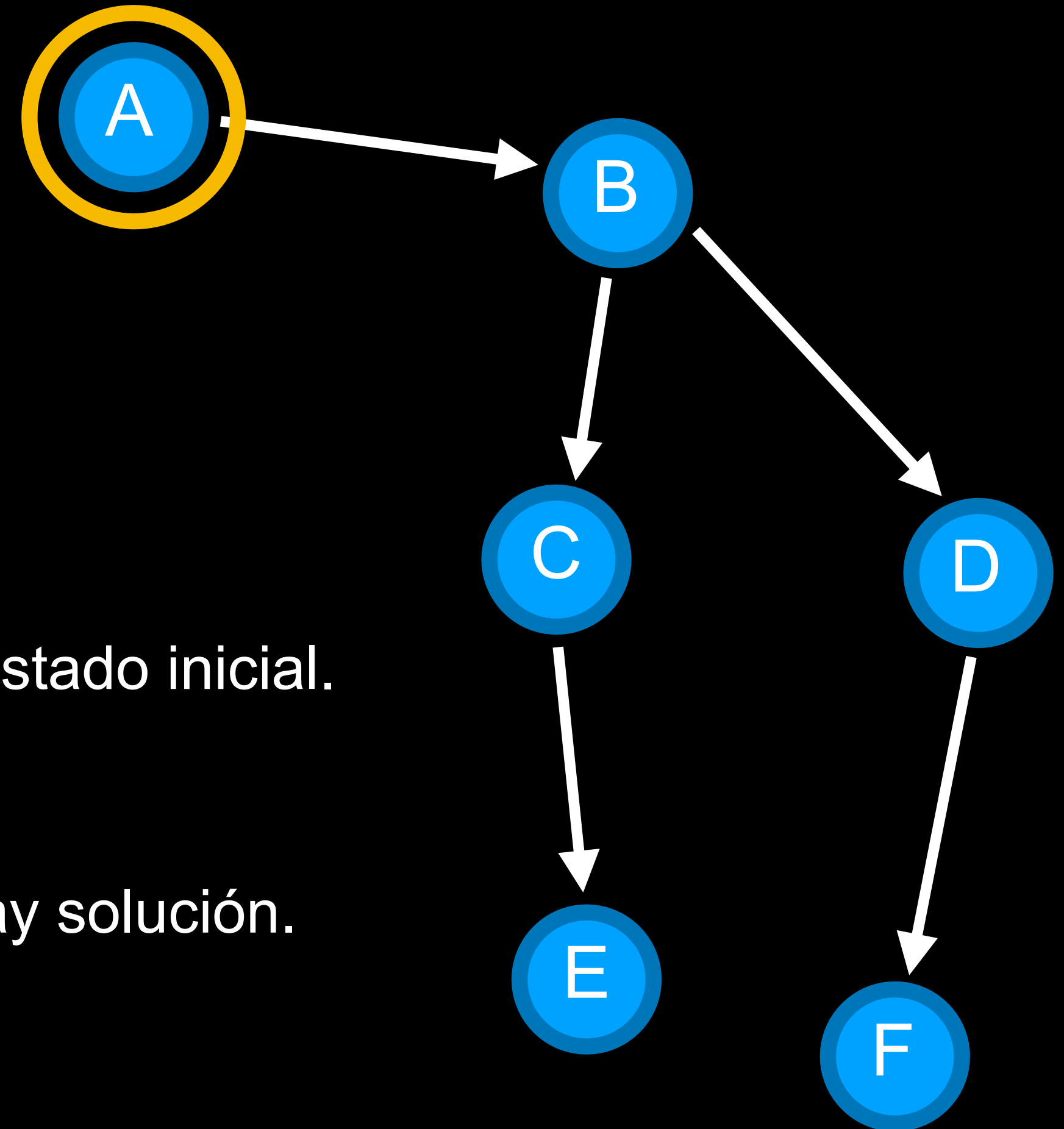
Ruta desde A hacia E?.

Frontera

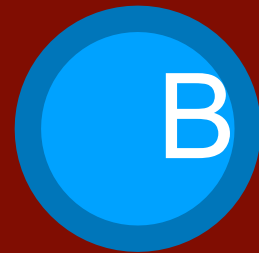


- Comience con una frontera que contiene el estado inicial.
- Repetir:
 - Si la frontera es vacía entonces, no hay solución.
 - **Remover un nodo de la frontera.**
 - Si el nodo contiene el estado meta, retornar la solución.
- **Expandir** nodo, añadir nodos resultantes a la frontera.

Ruta desde A hacia E?.



Frontera

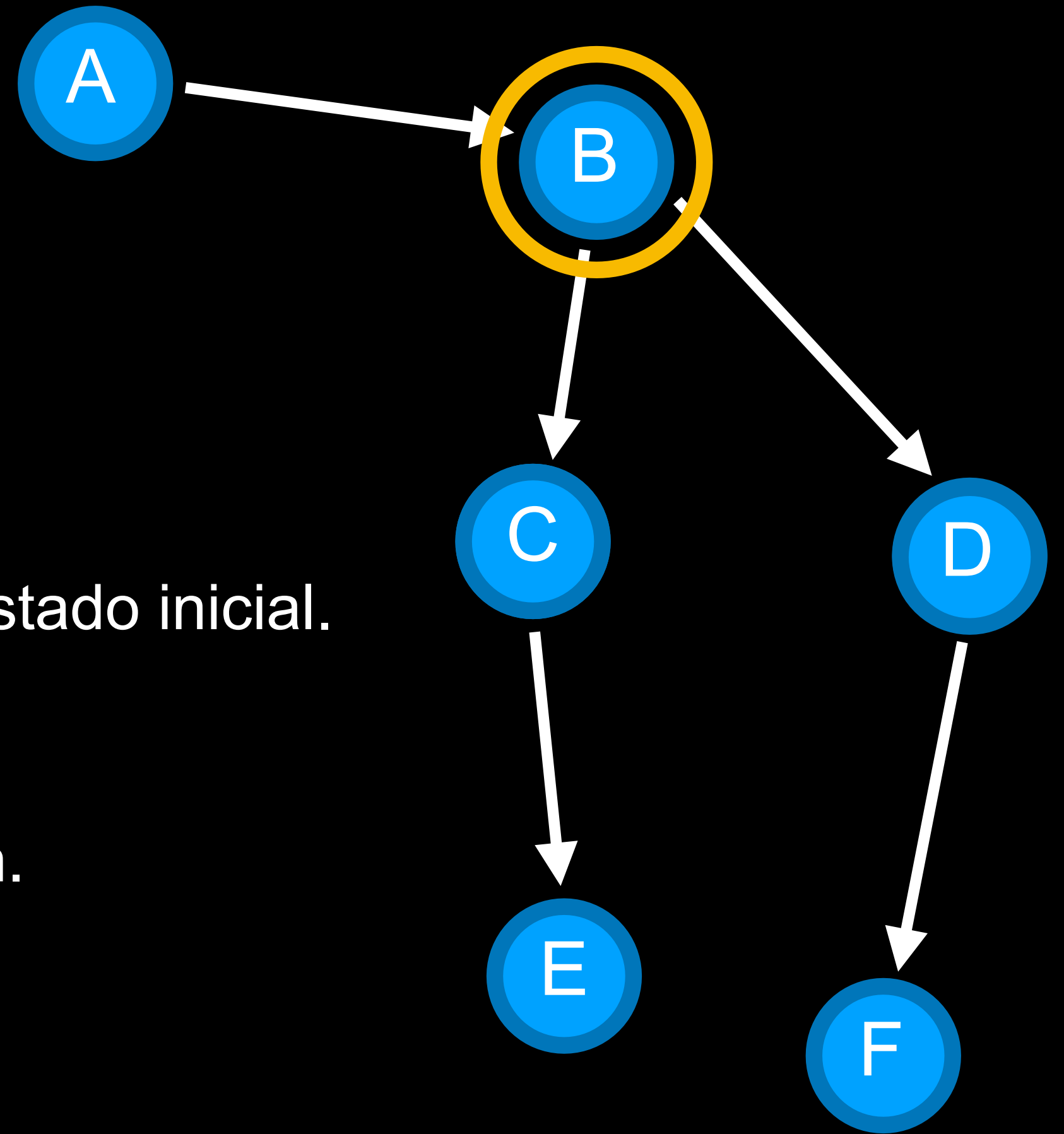


- Comience con una frontera que contiene el estado inicial.
- Repetir:
 - Si la frontera está vacía, entonces no hay solución.
 - Remover un nodo de la frontera.
 - Si el nodo es la meta, retorne la solución.
 - **Expandir nodo**, añadir nodos resultantes a la frontera.

Ruta desde A hacia E?.

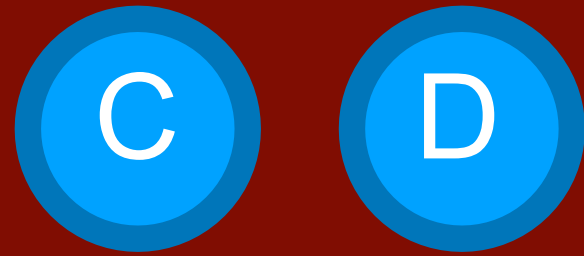
Frontera

- Comience con una **frontera** que contiene el estado inicial.
- Repetir:
 - Si el nodo frontera está vacío, sin solución.
 - **Remover nodo de la frontera**
 - Si el nodo es la meta, retorne la solución.
 - **Expandir** nodo, añadir nodos resutantes a la frontera.

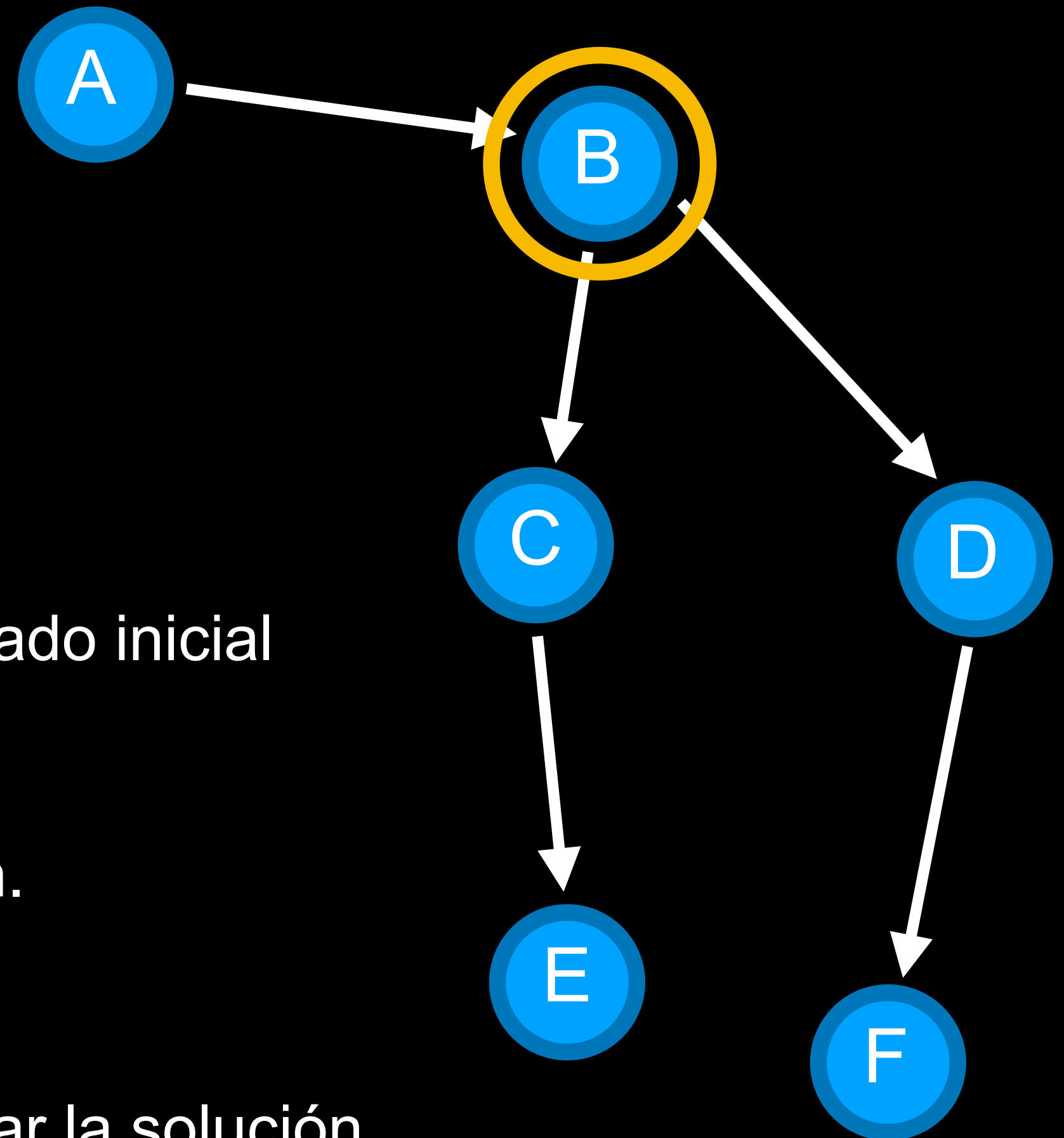


Ruta desde A hacia E?.

Frontier



- Comience con una frontera que contiene el estado inicial
- Repetir:
 - Si el nodo frontera está vacío, sin solución.
 - Remover el nodo de la frontera.
 - Si el nodo contiene la meta estado, retornar la solución.
 - **Expandir** nodo, adicionar nodos resultantes a la frontera.

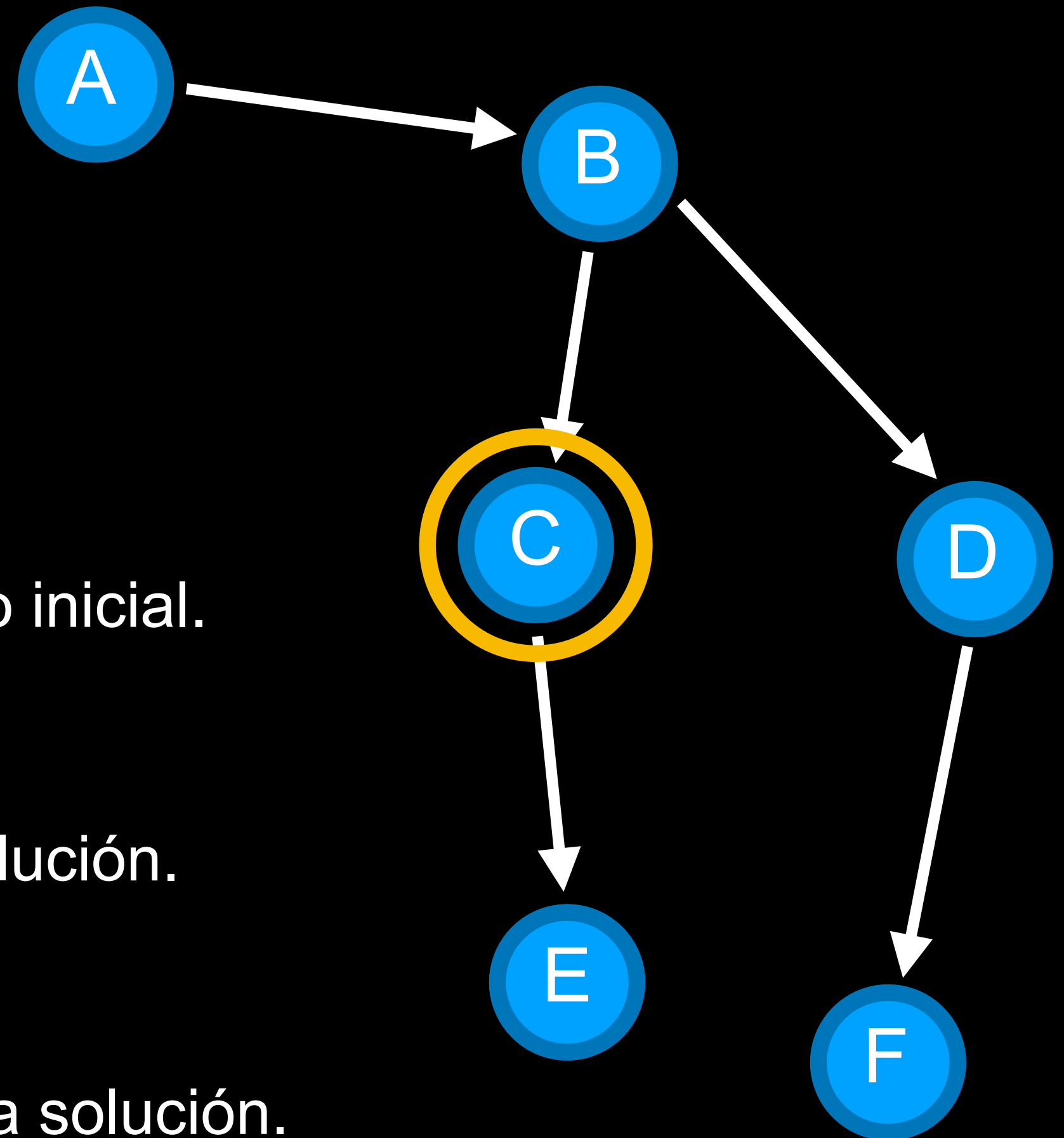


Ruta desde A hacia E?.

Frontera

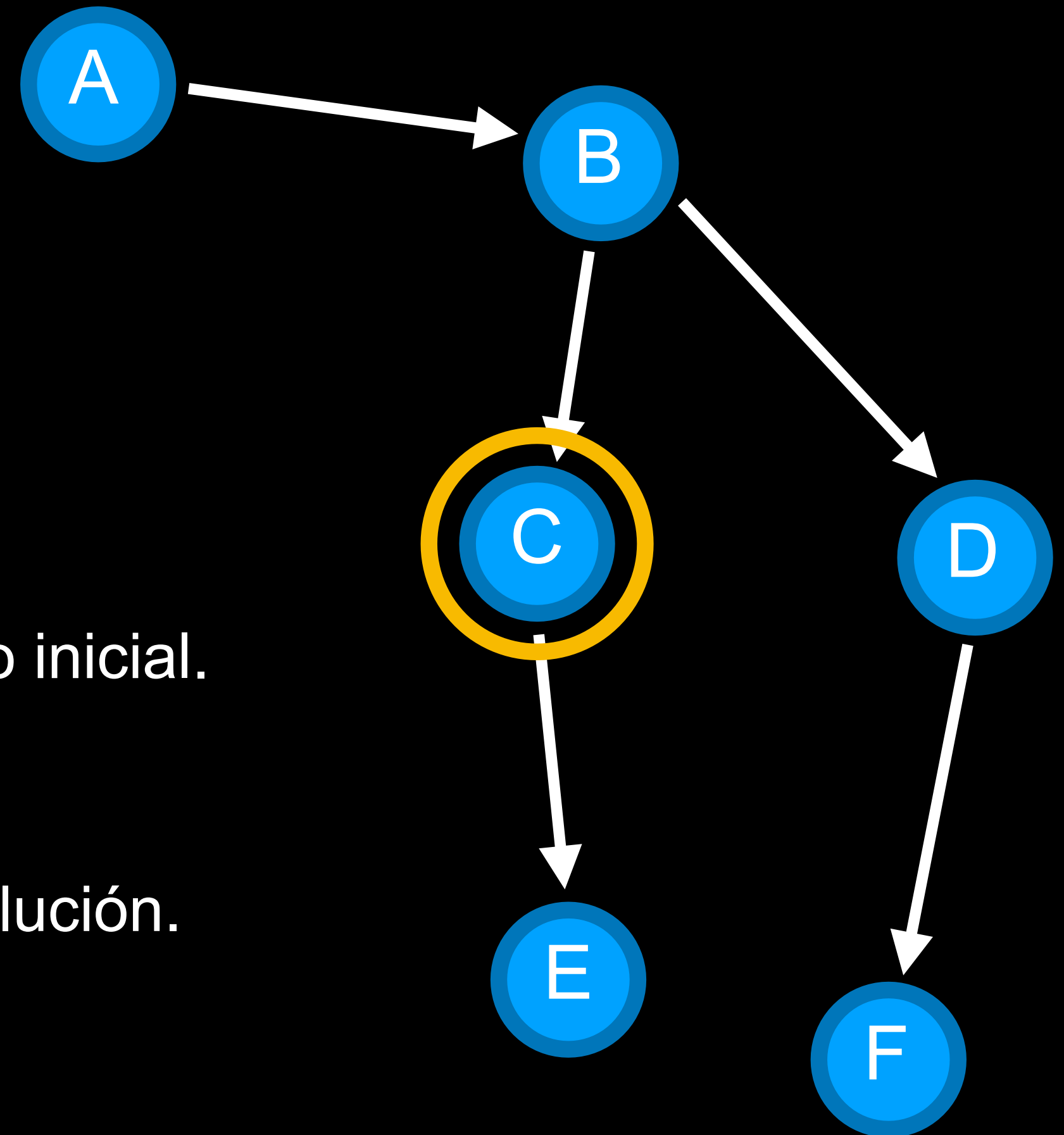
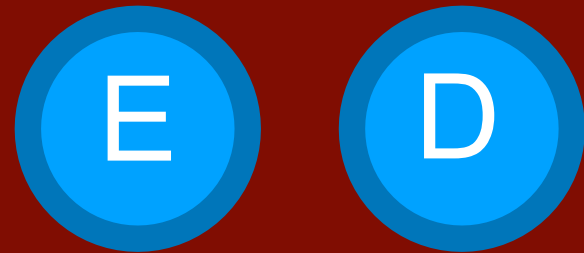


- Comience con la **frontera** que contiene el estado inicial.
- Repetir:
 - Si la frontera está vacía, entonces no hay solución.
 - **Remover nodo de la frontera.**
 - Si el nodo contiene la meta estado, retorne la solución.
 - **Expanda** nodo, adicione nodos resultantes a la frontera.



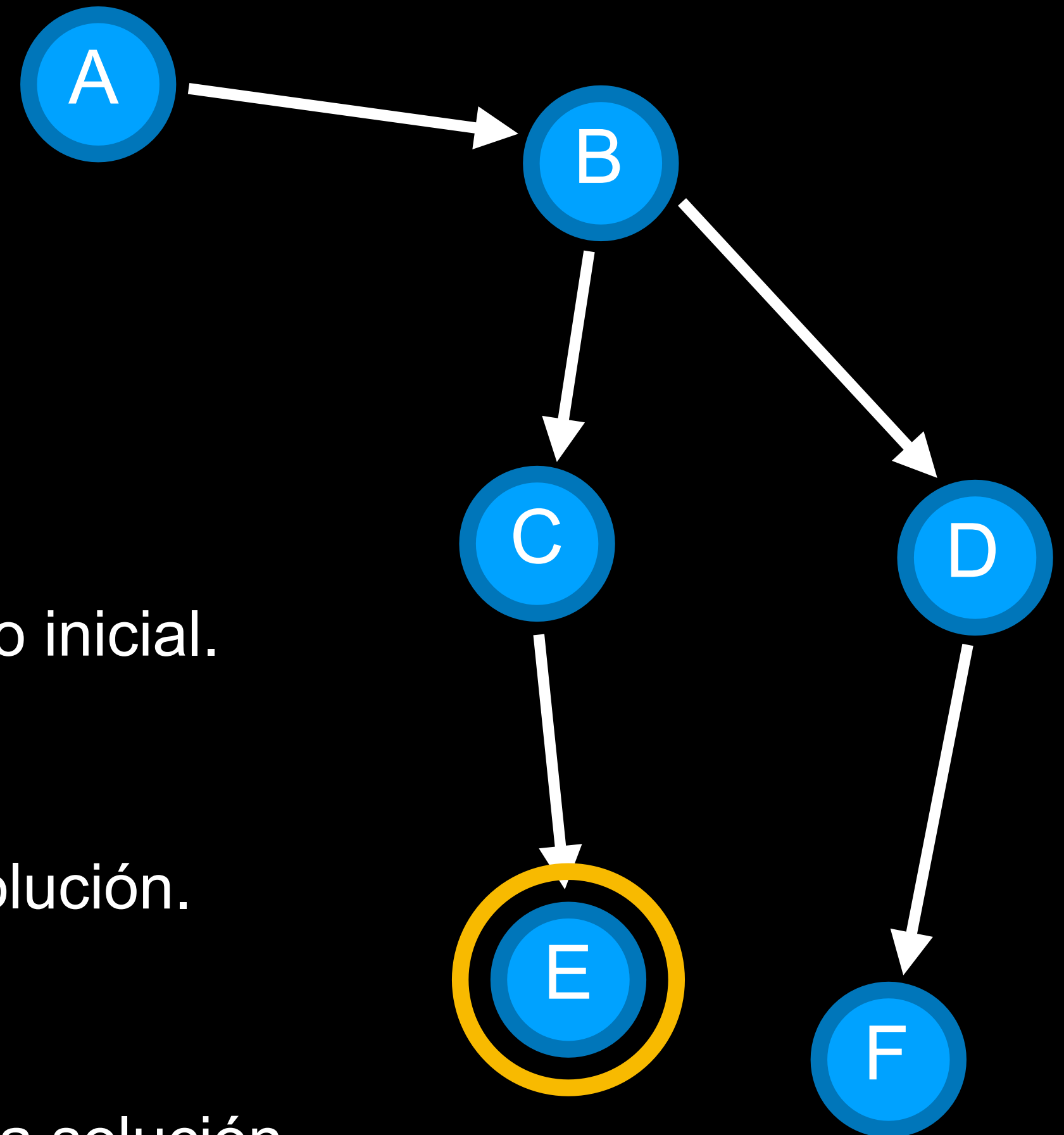
Ruta desde A hacia E?.

Frontier



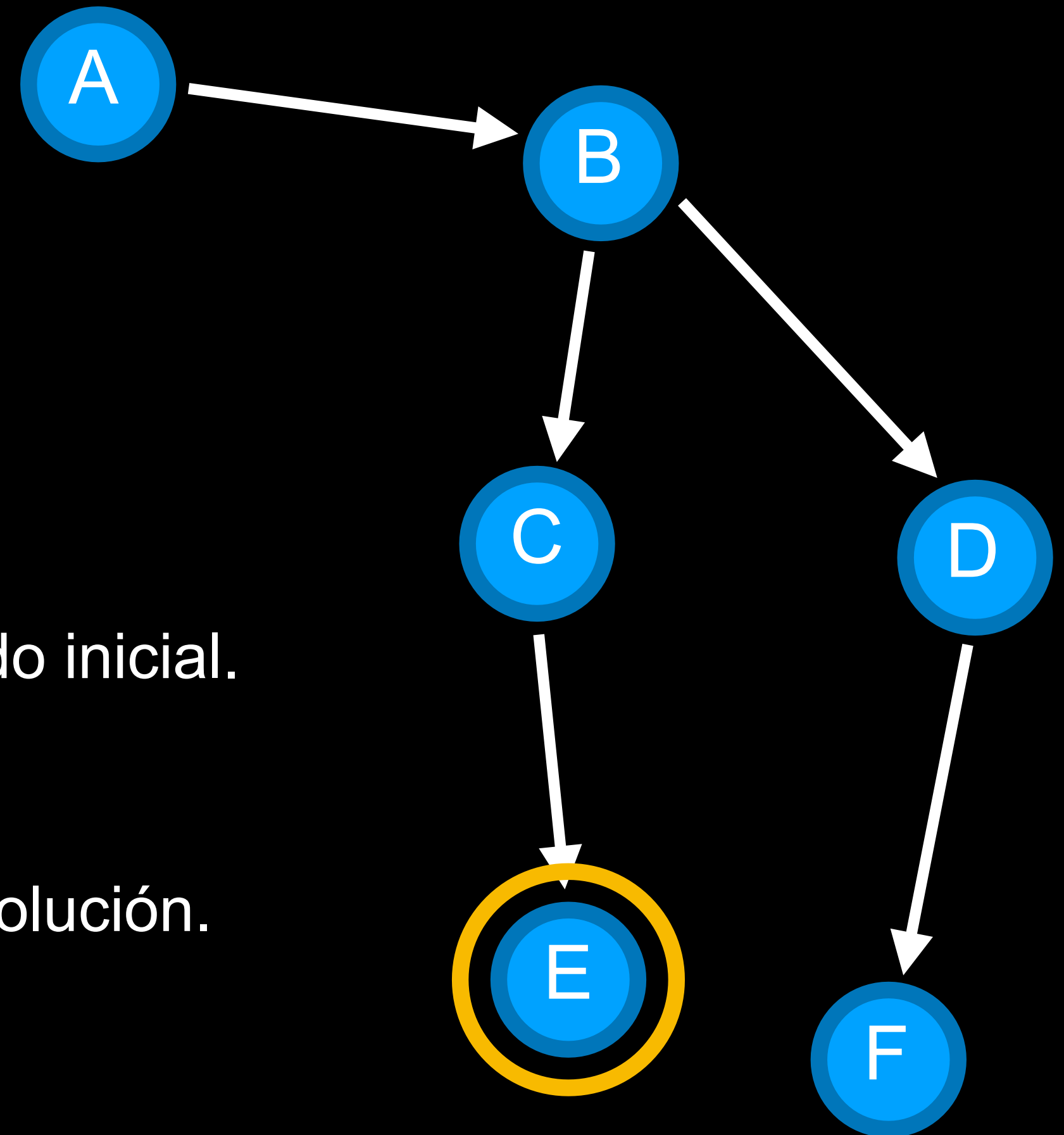
- Comience con la **frontera** que contiene el estado inicial.
- Repetir:
 - Si la frontera está vacía, entonces no hay solución.
 - Remover nodo de la frontera.
 - Si el nodo contiene la meta estado, retorne la solución.
 - **Expanda** nodo, adicione nodos resultantes a la frontera.

Ruta desde A hacia E?.



- Comience con la **frontera** que contiene el estado inicial.
- Repetir:
 - Si la frontera está vacía, entonces no hay solución.
 - **Remover** nodo de la frontera.
 - Si el nodo contiene la meta estado, retorne la solución.
 - **Expand** nodo, adicione nodos resultantes a la frontera.

Ruta desde A hacia E?.

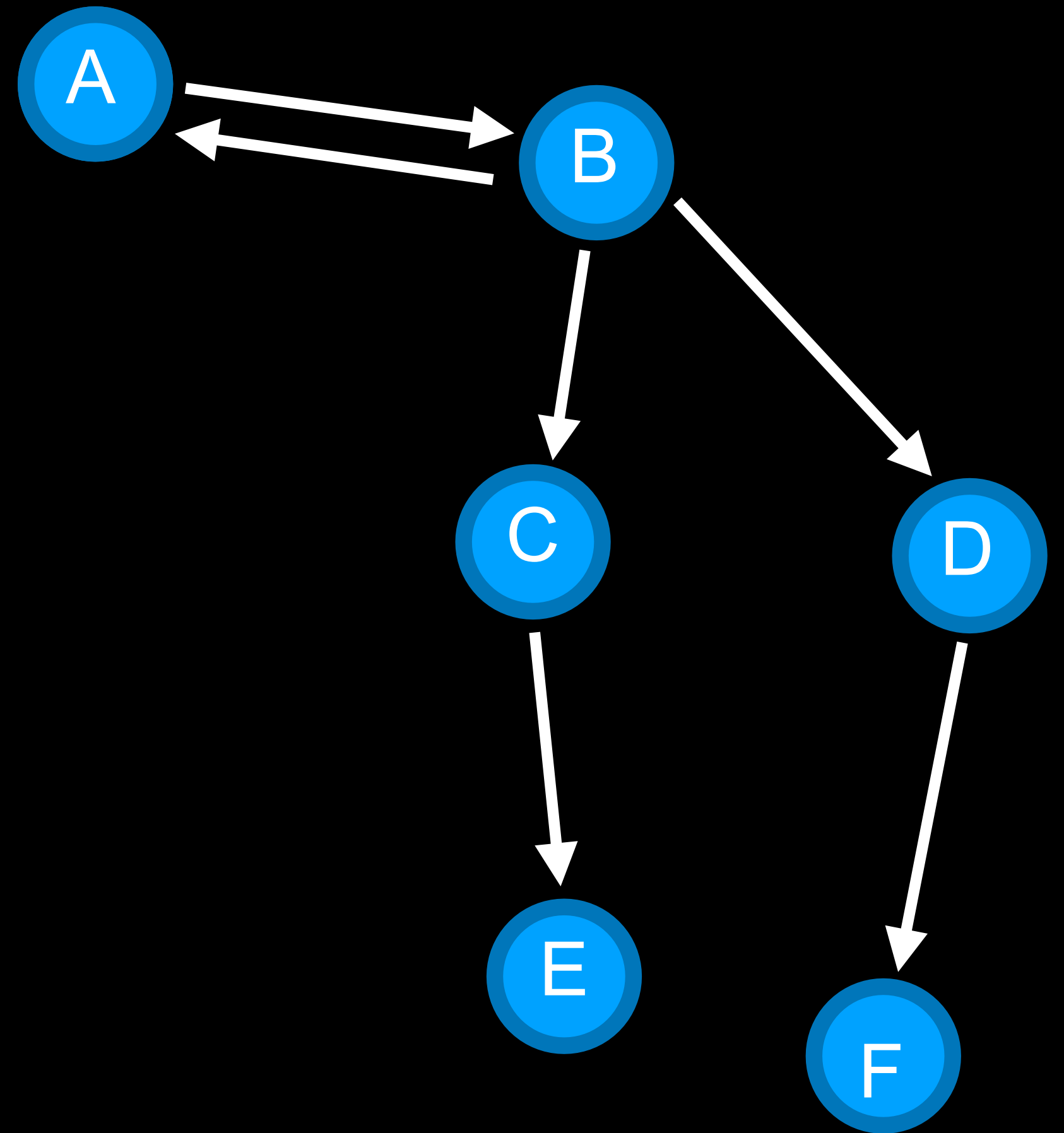


- Comience con la **frontera** que contiene el estado inicial.
- Repetir:
 - Si la frontera está vacía, entonces no hay solución.
 - Remover nodo de la frontera.
 - Si el nodo contiene la meta estado, retorne la solución.
 - **Expanda** nodo, adicione nodos resultantes a la frontera.

Que podría salir mal?

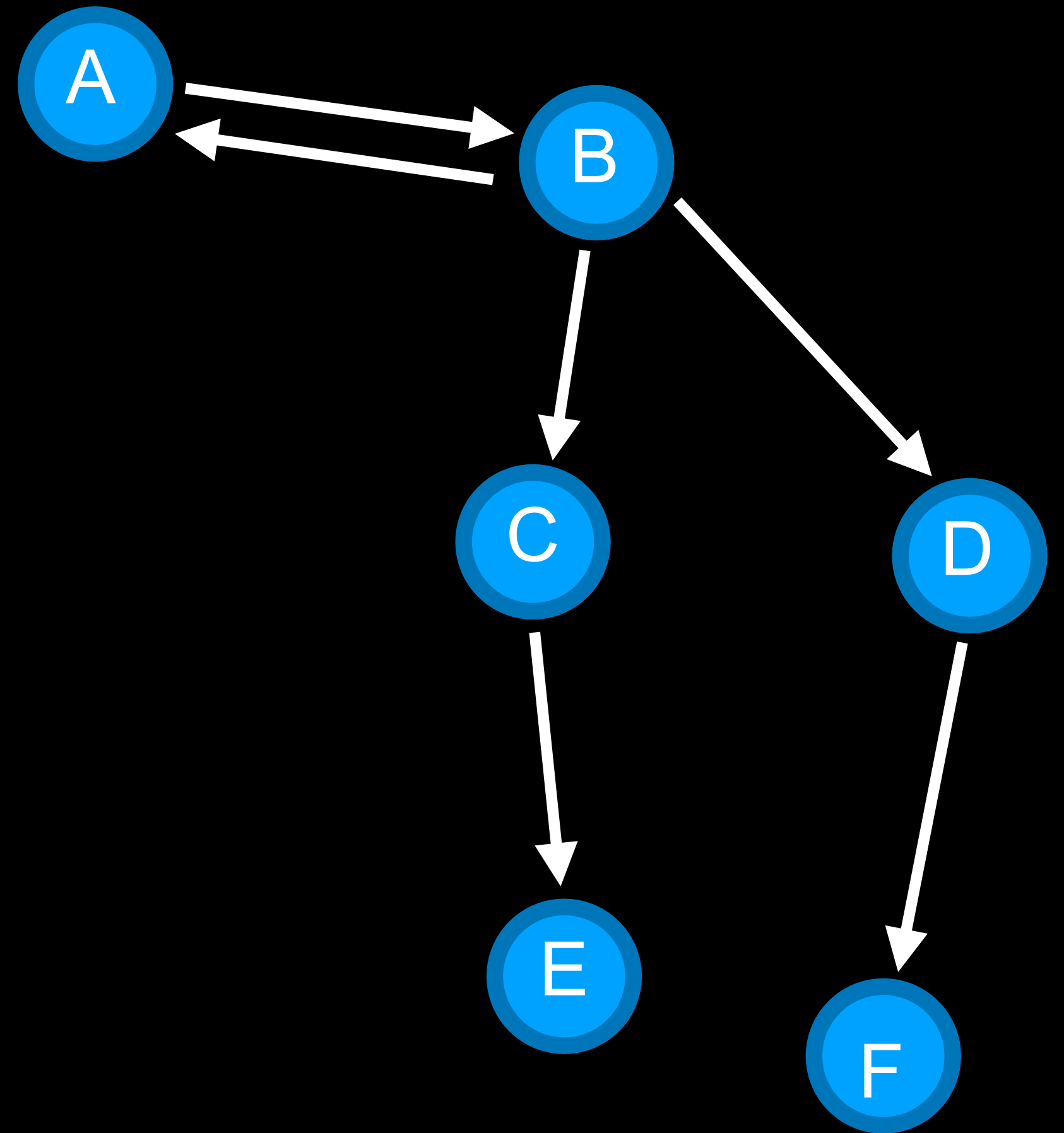
Ruta desde A hacia E?.

Frontera



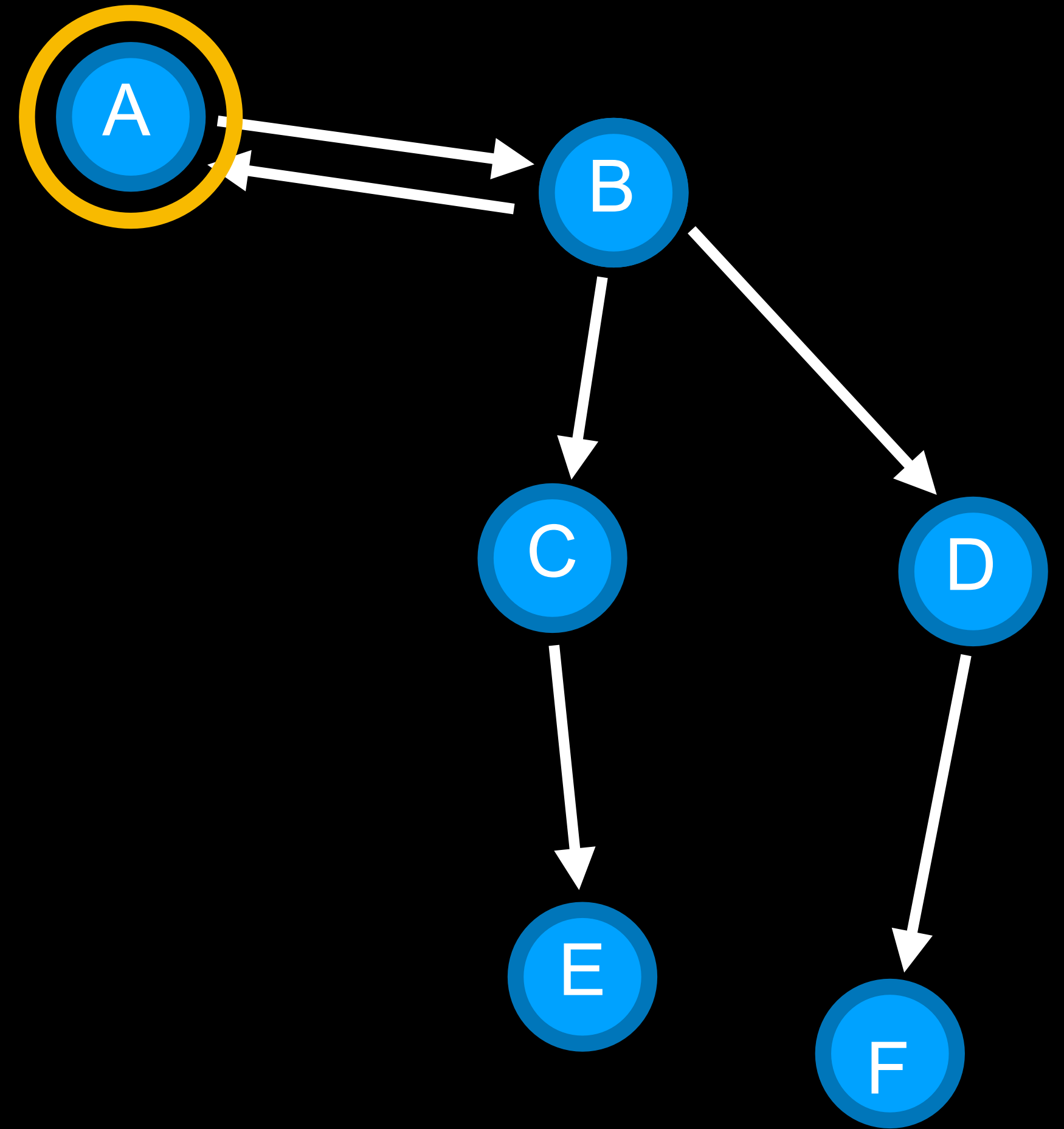
Ruta desde A hacia E?.

Frontera



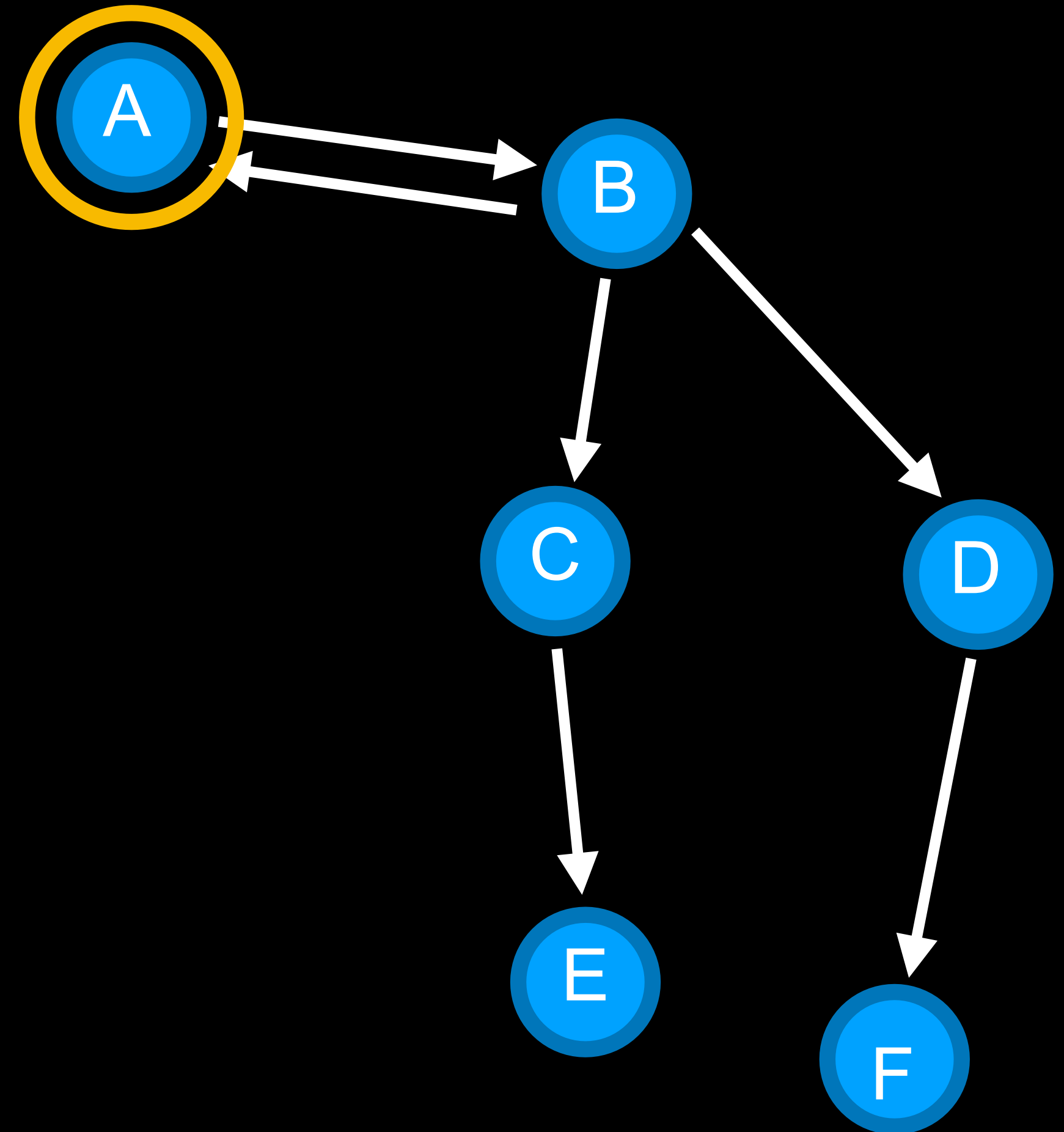
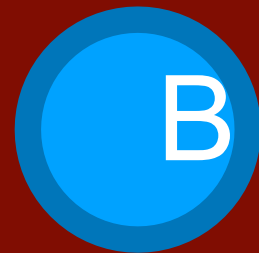
Ruta desde A hacia E?.

Frontera



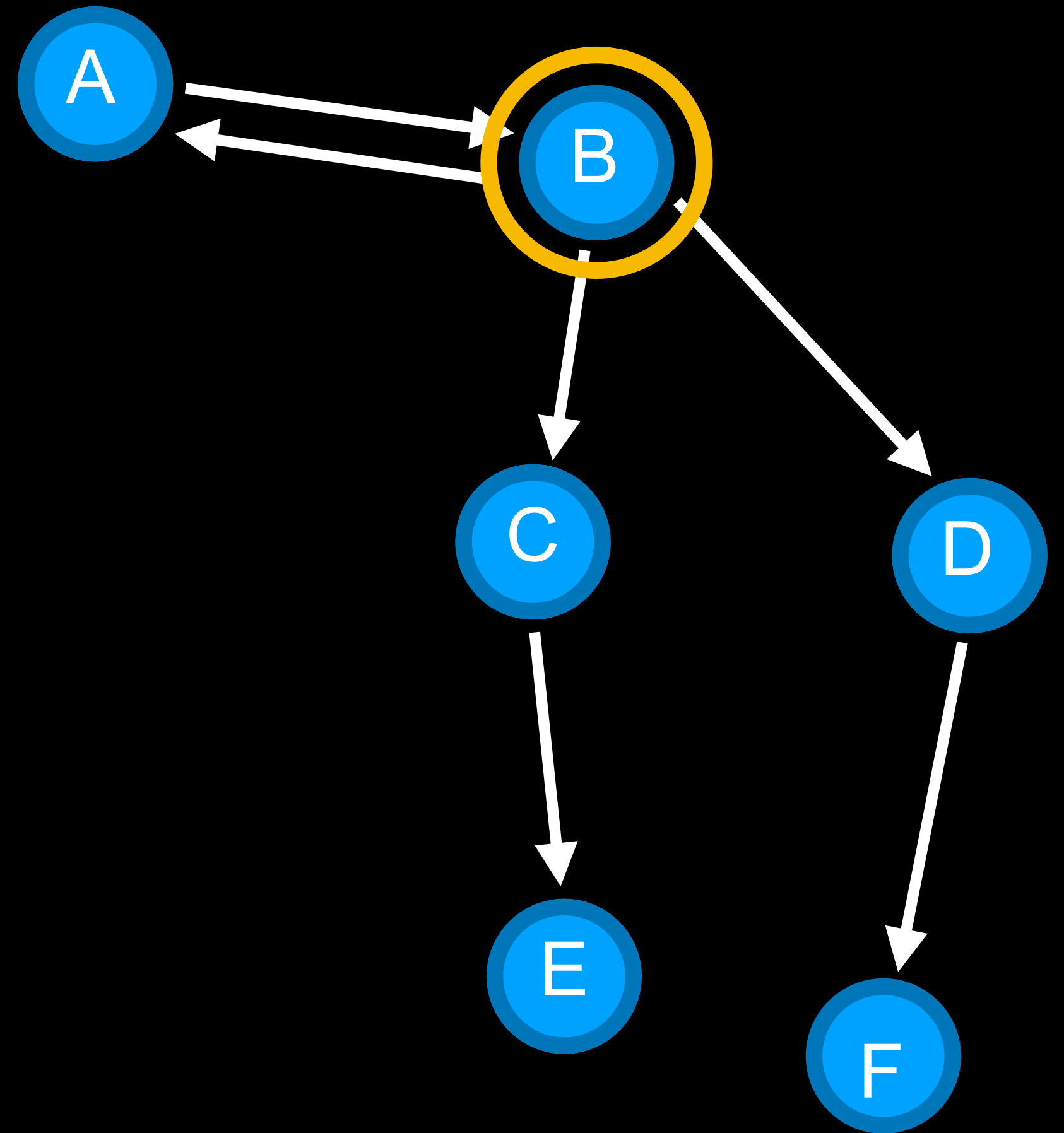
Ruta desde A hacia E?.

Frontera



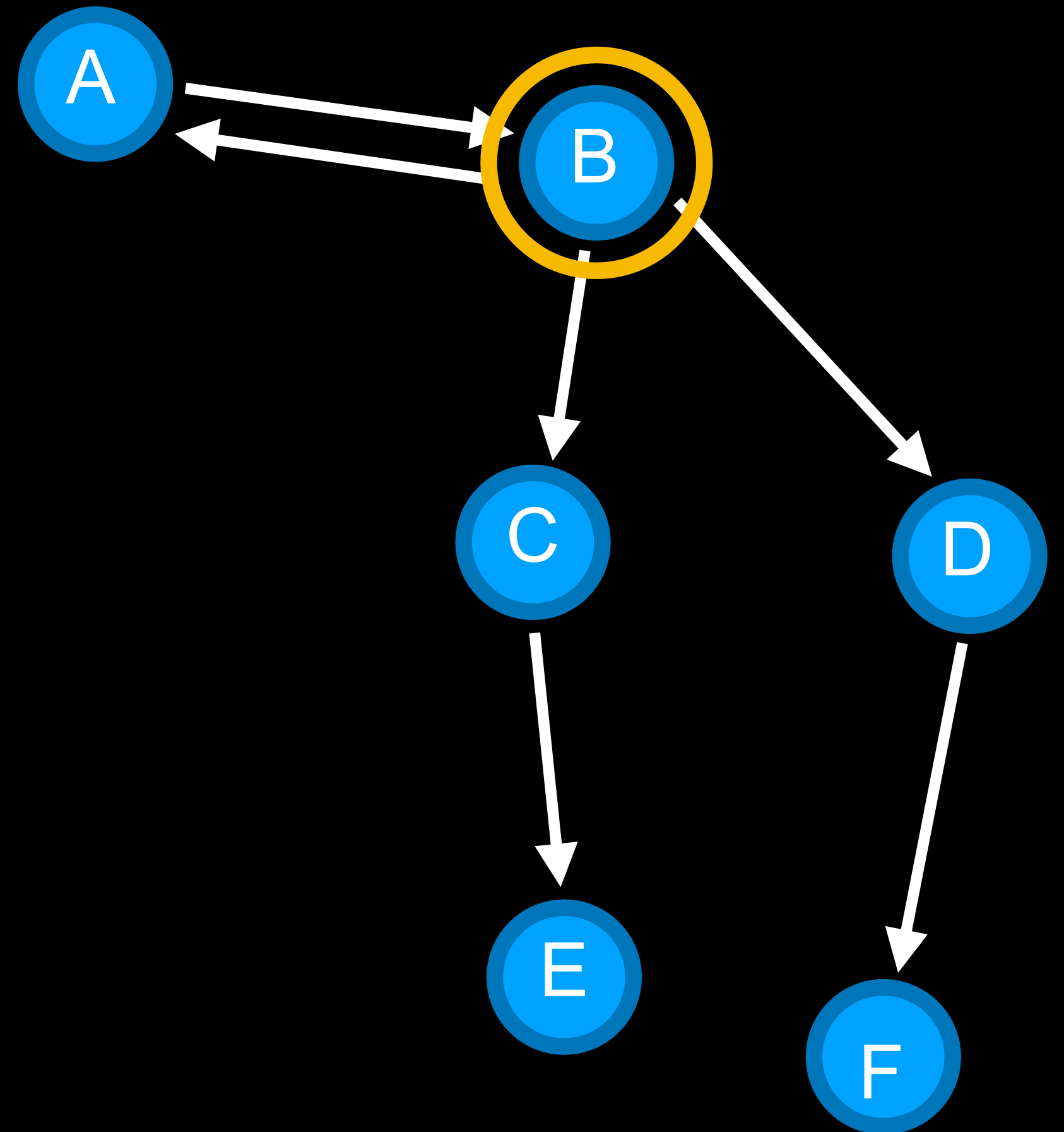
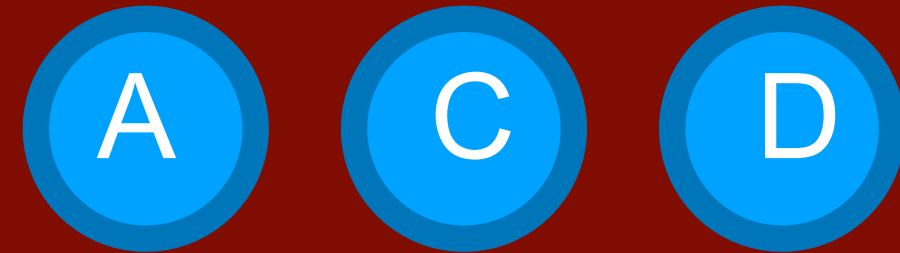
Ruta desde A hacia E?.

Frontera



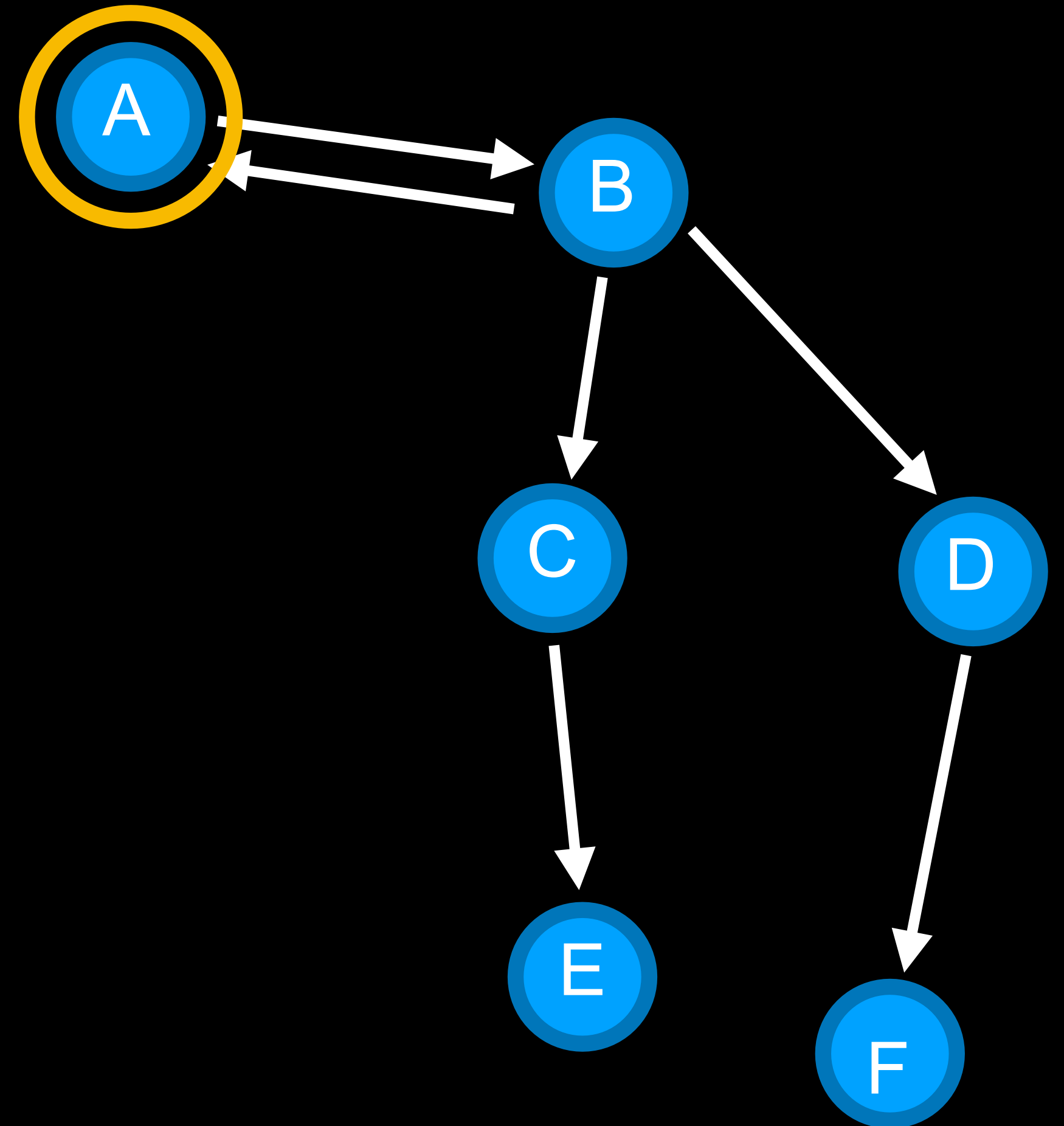
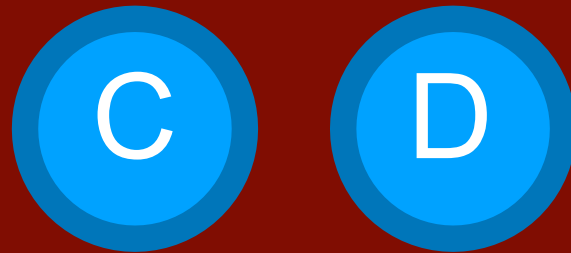
Ruta desde A hacia E?.

Frontier



Ruta desde A hacia E?.

Frontier



Aproximación revisada

- Comience con una **frontera que** contenga el estado inicial.
- Comenzar con un **conjunto explorado** vacío.
- Repetir:
 - Si la frontera está vacía entonces, no hay solución
 - Remover un nodo de la frontera.
 - Si el nodo contiene el estado meta, retornar la solución.
 - Adicionar el nodo al conjunto explorado.
 - **Expandir** nodo, adicionar nodos resultantes a la frontera, si aún no están en la frontera o el conjunto explorado

Aproximación revisada

- Comience con una **frontera que** contenga el estado inicial.
- Comenzar con un **conjunto explorado** vacío.
- Repetir:
 - Si la frontera está vacía entonces, no hay solución
 - **Remover un nodo de la frontera.**
 - Si el nodo contiene el estado meta, retornar la solución.
 - Adicionar el nodo al conjunto explorado.
 - **Expandir** nodo, adicionar nodos resultantes a la frontera, si aún no están en la frontera o el conjunto explorado

Pila LIFO

Del tipo de datos

last-in first-out

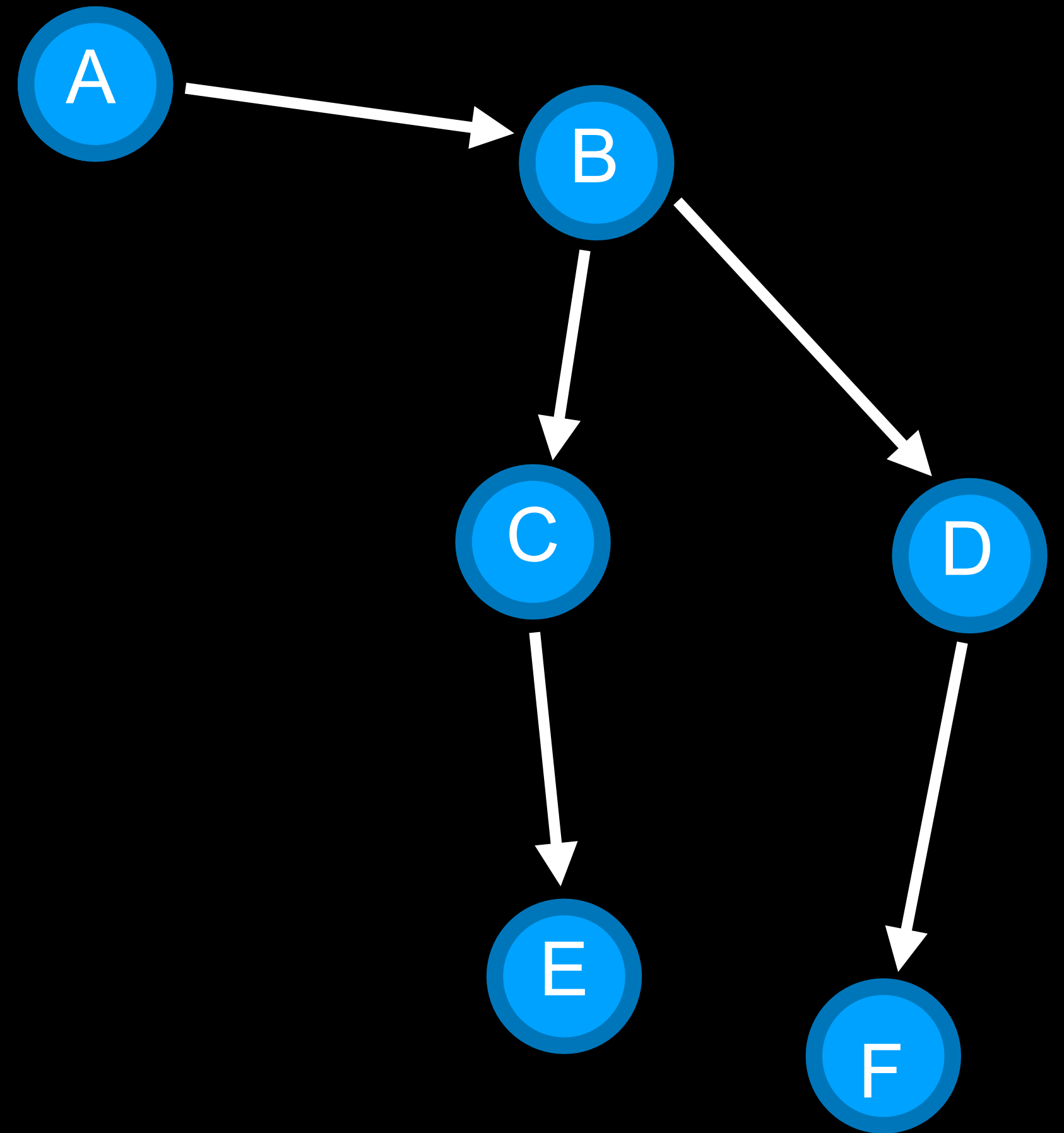
Último en llegar, primero en salir

línea 127 StackFrontier

Ruta desde A hacia E?.

Frontera

Conjunto explorado

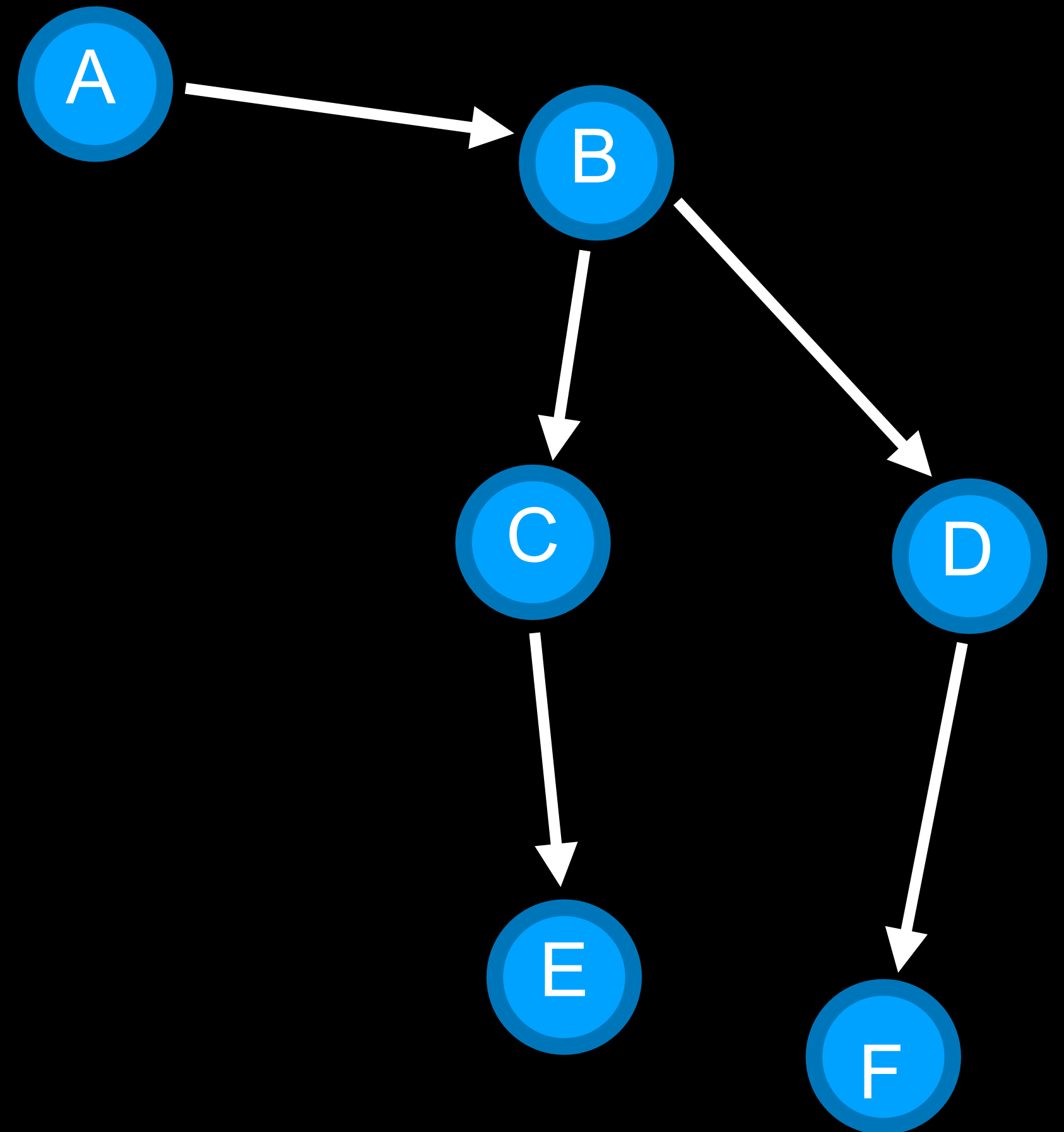


Ruta desde A hacia E?.

Frontera



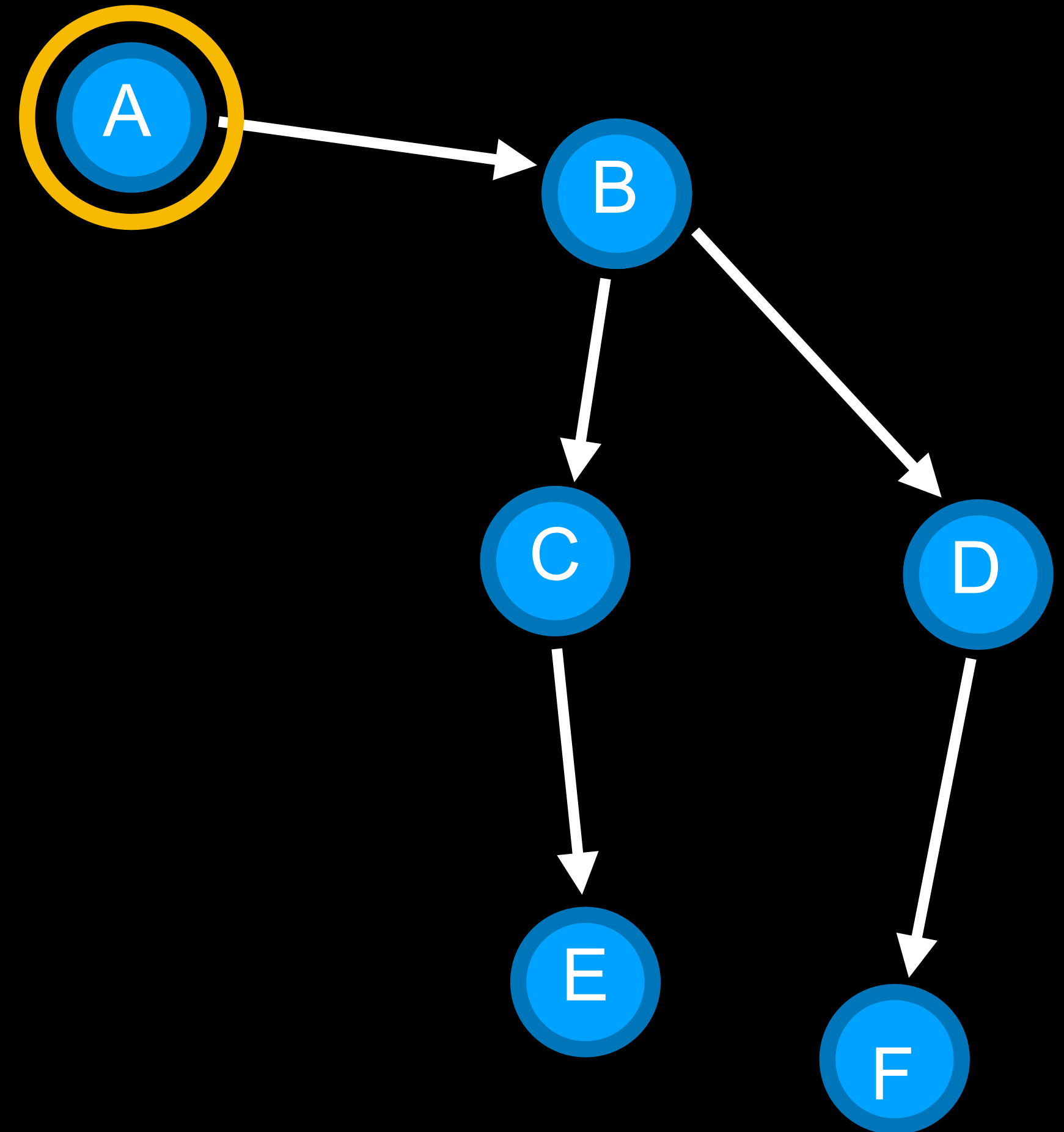
Conjunto explorado



Ruta desde A hacia E?.

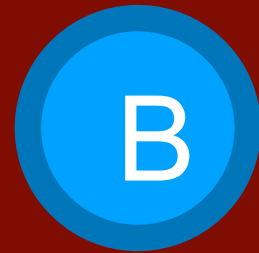
Frontera

Conjunto explorado

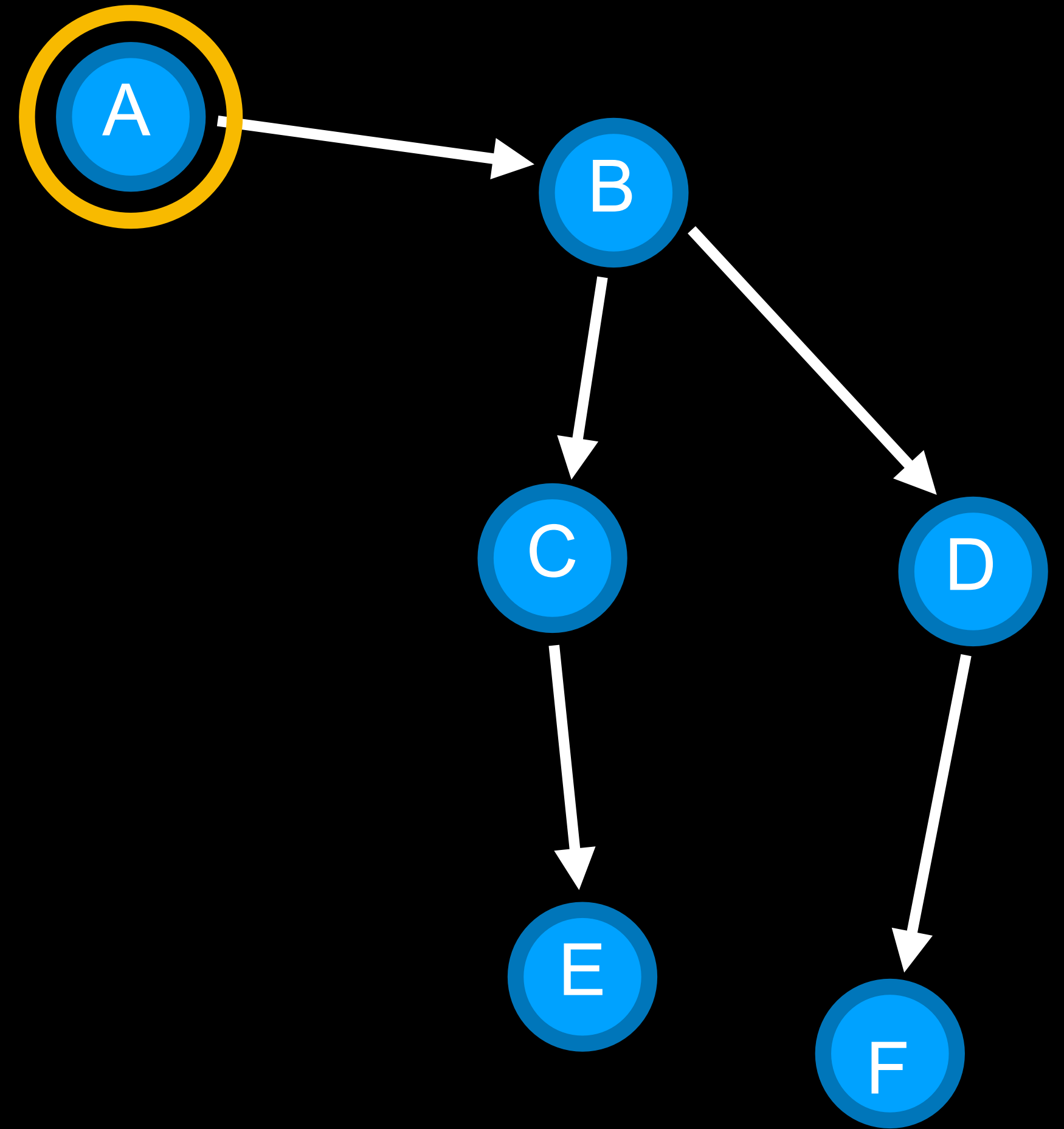


Ruta desde A hacia E?.

Frontera

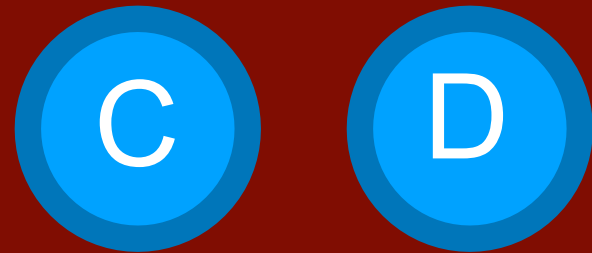


Conjunto explorado

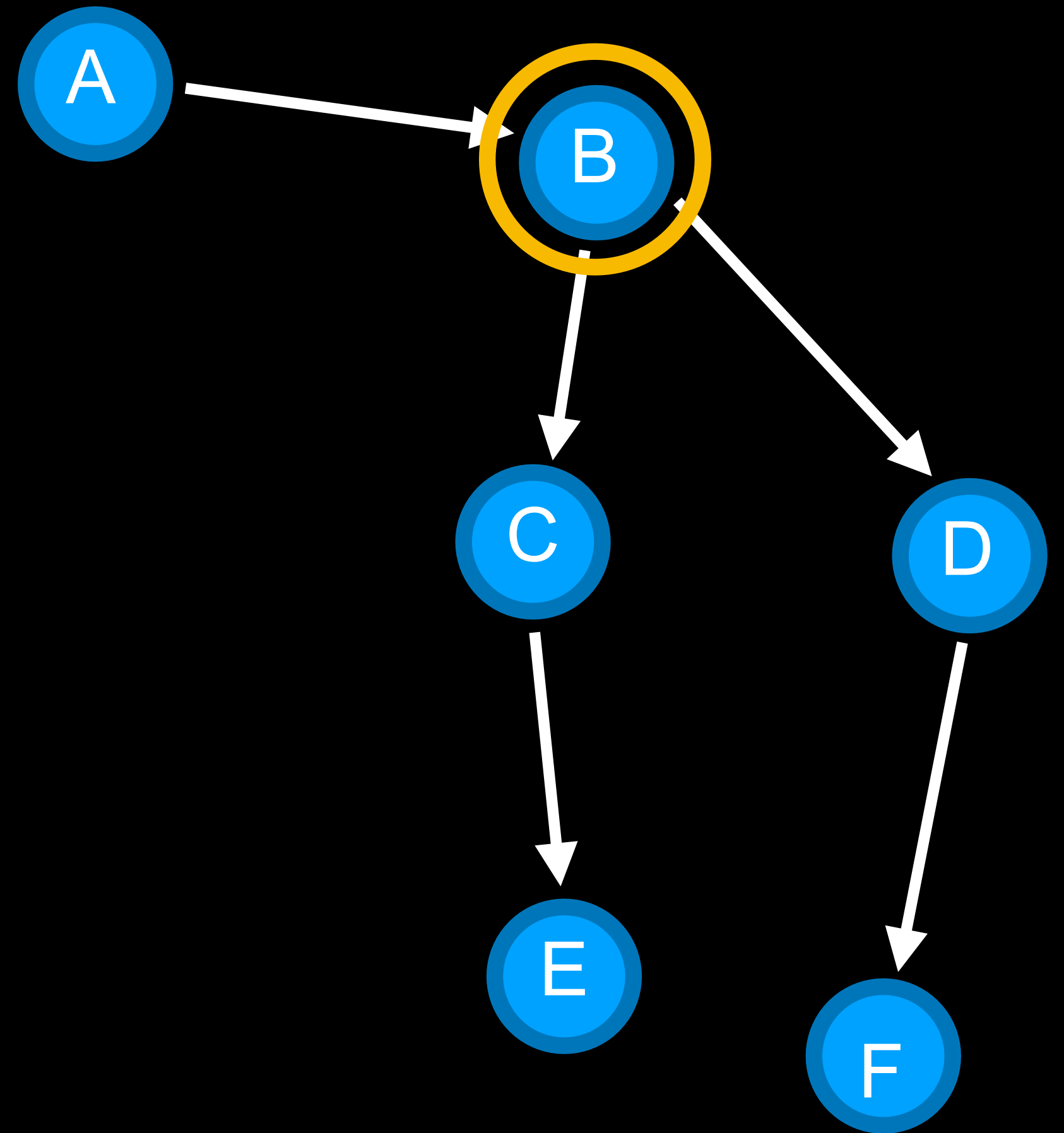
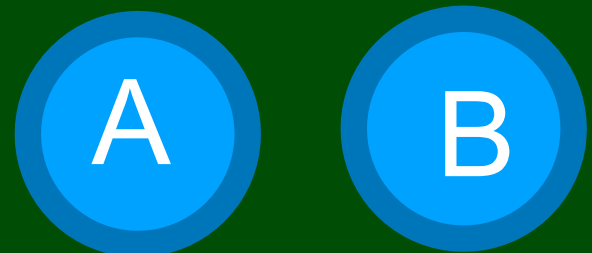


Ruta desde A hacia E?.

Frontera

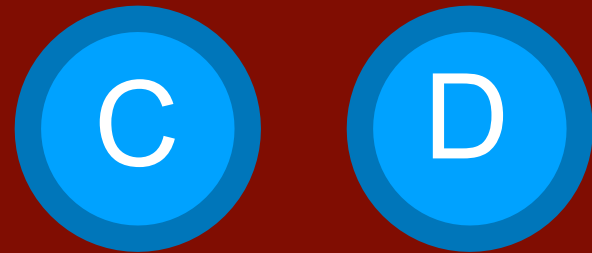


Conjunto explorado

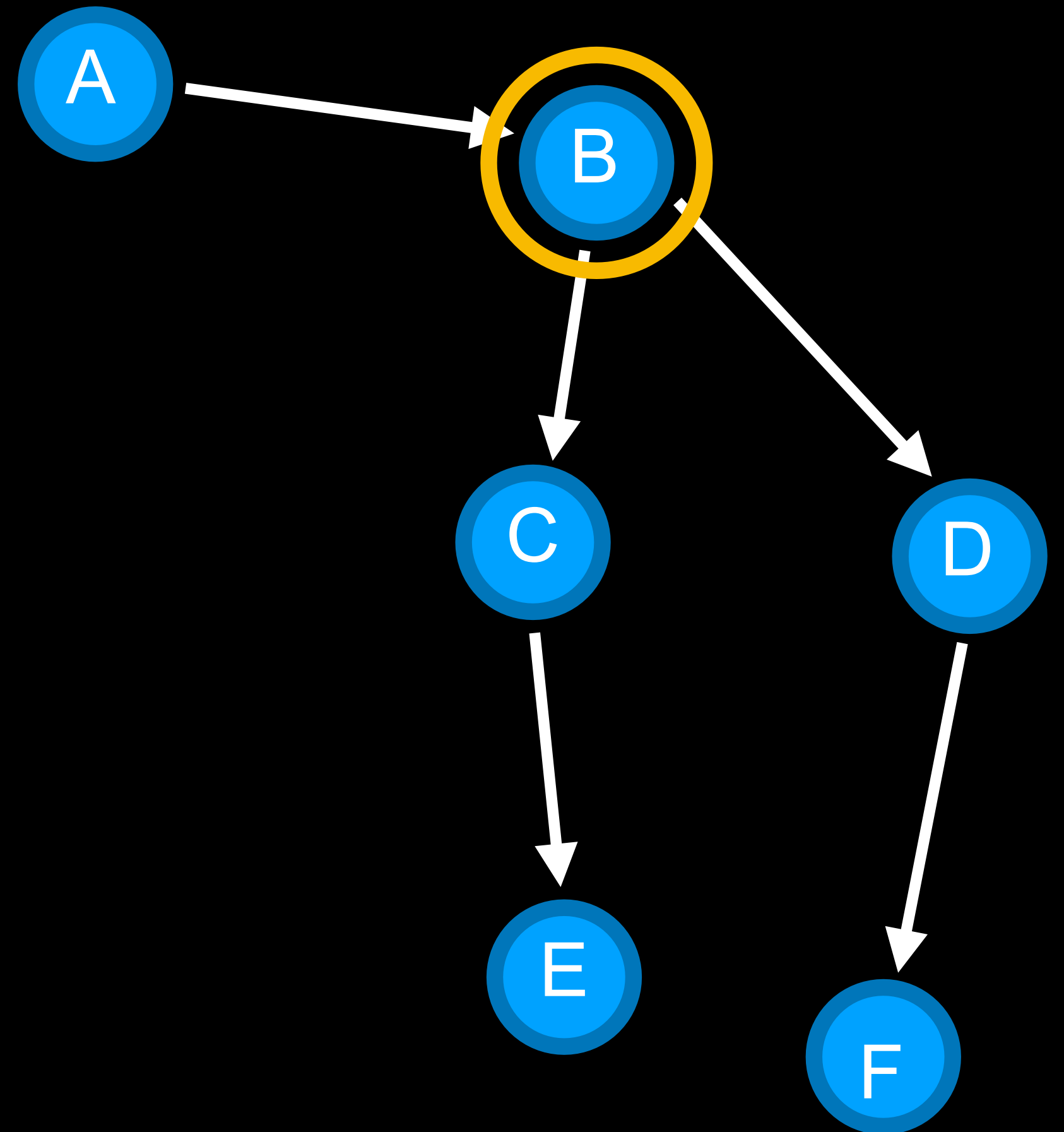
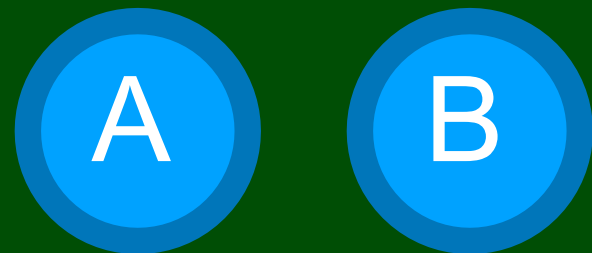


Ruta desde A hacia E?.

Frontera

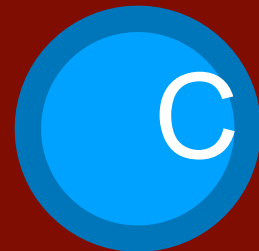


Conjunto Explorado

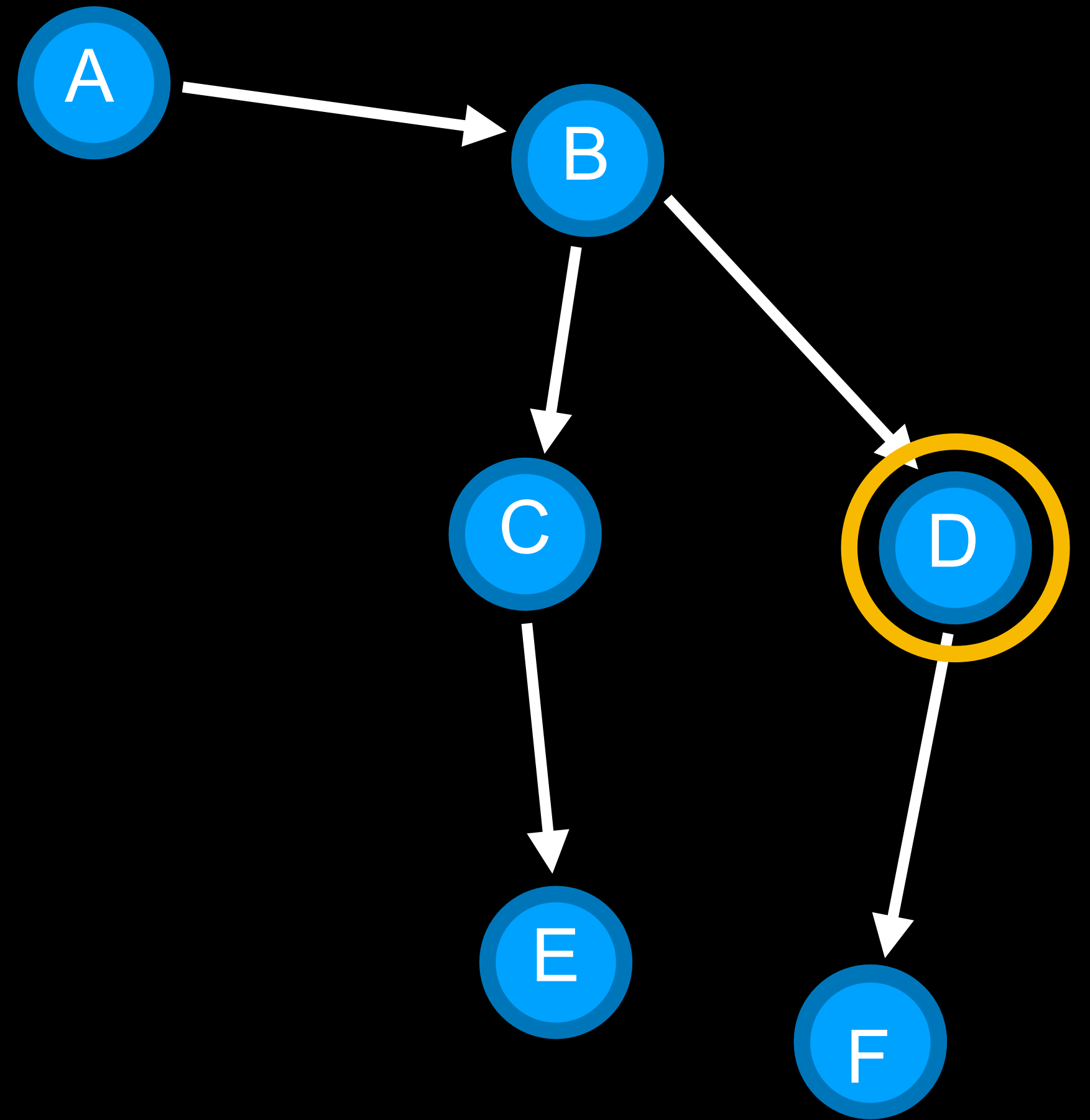


Camino de A a E?

Frontera

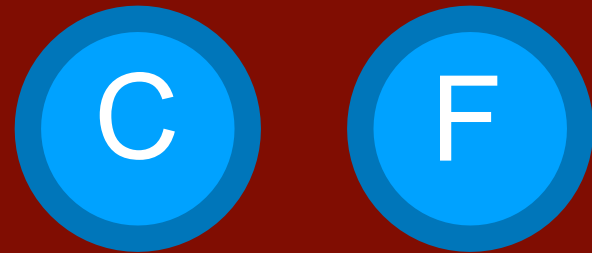


Conjunto explorado

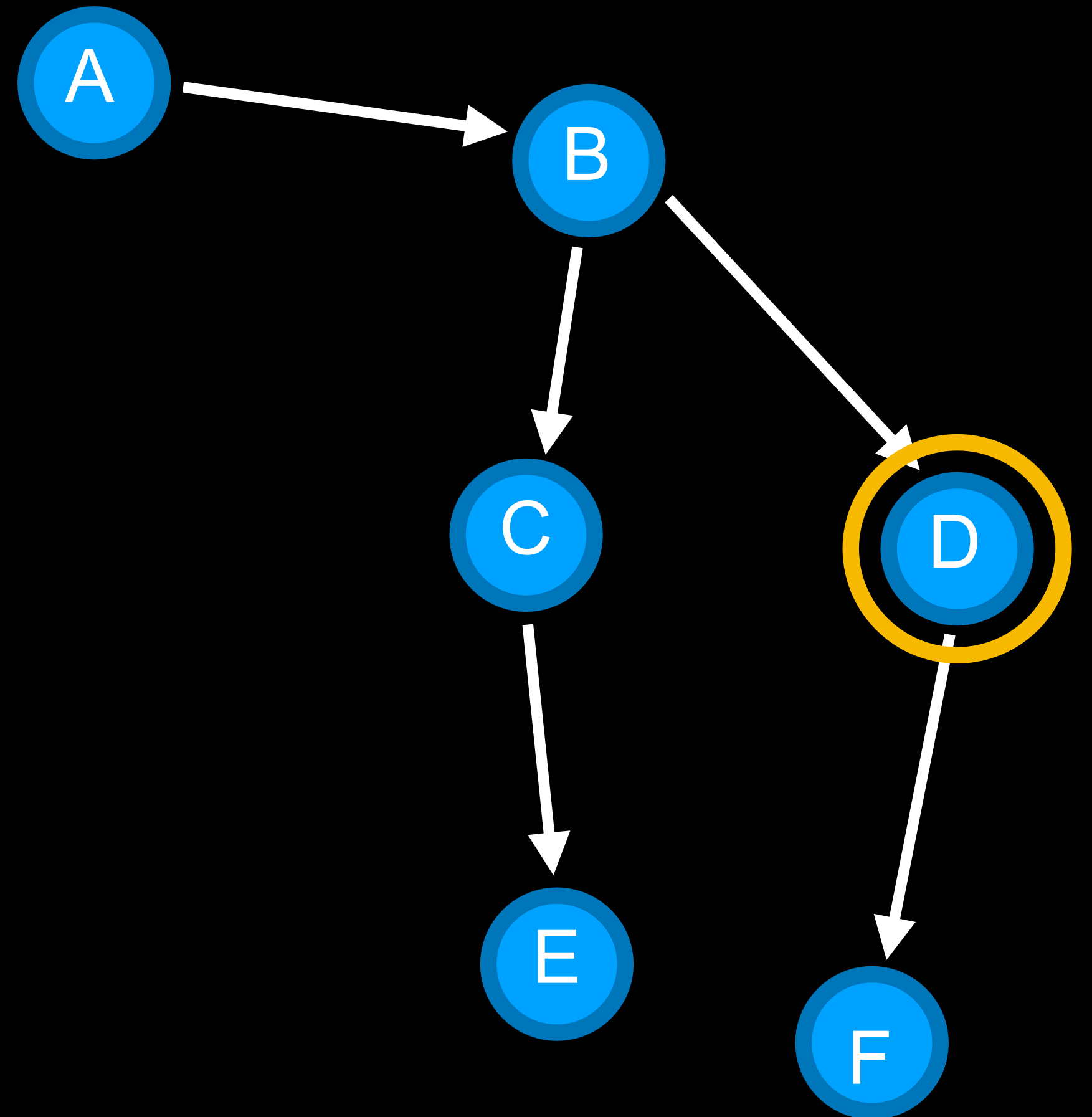


Camino de A a E?

Frontera

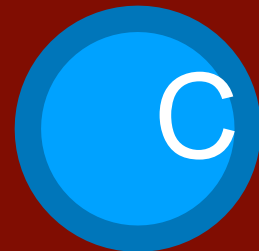


Conjunto explorado

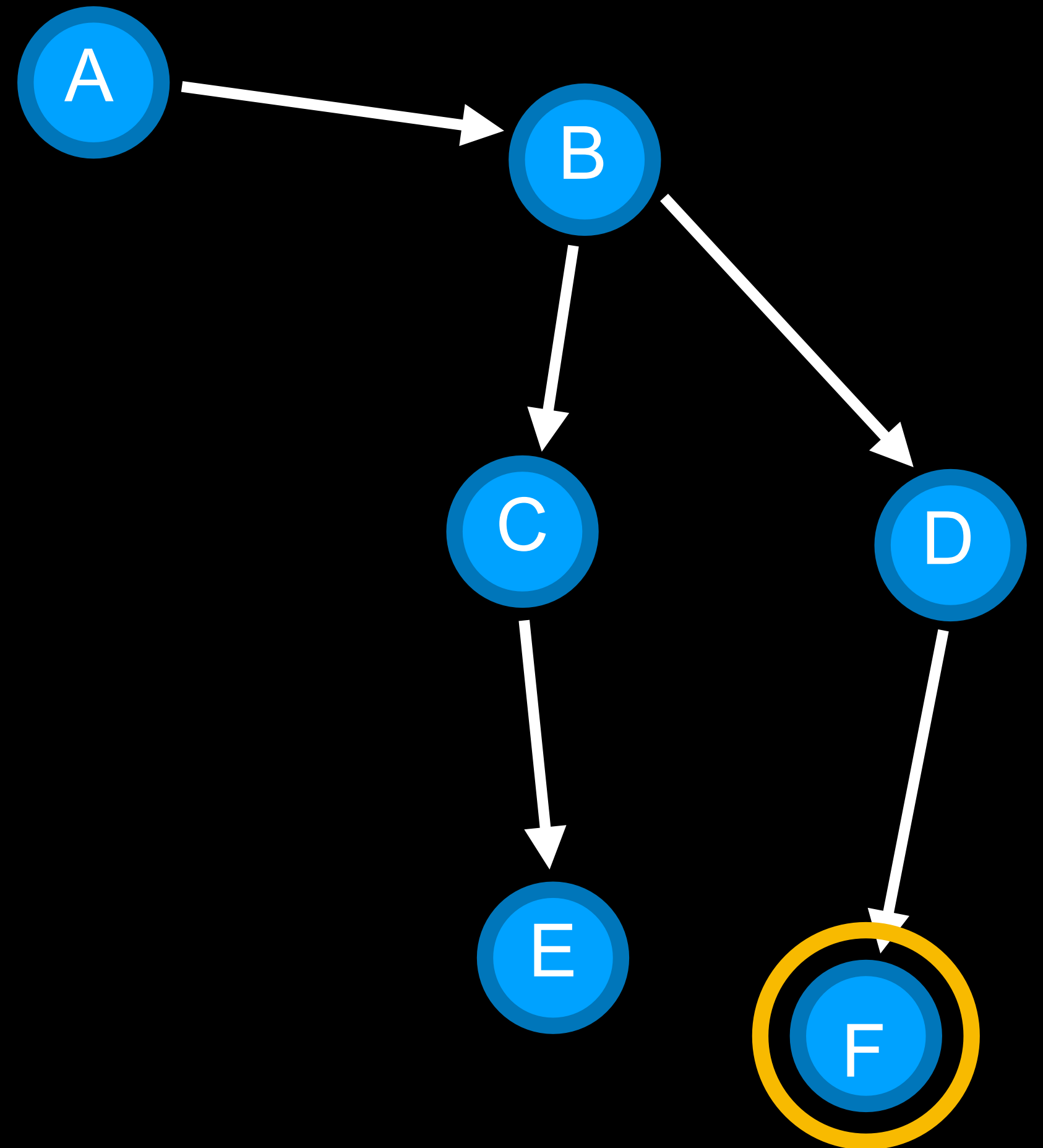
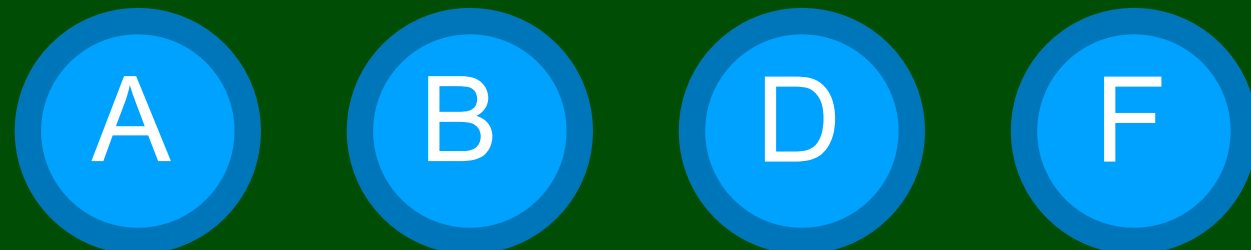


Camino de A a E?

Frontera



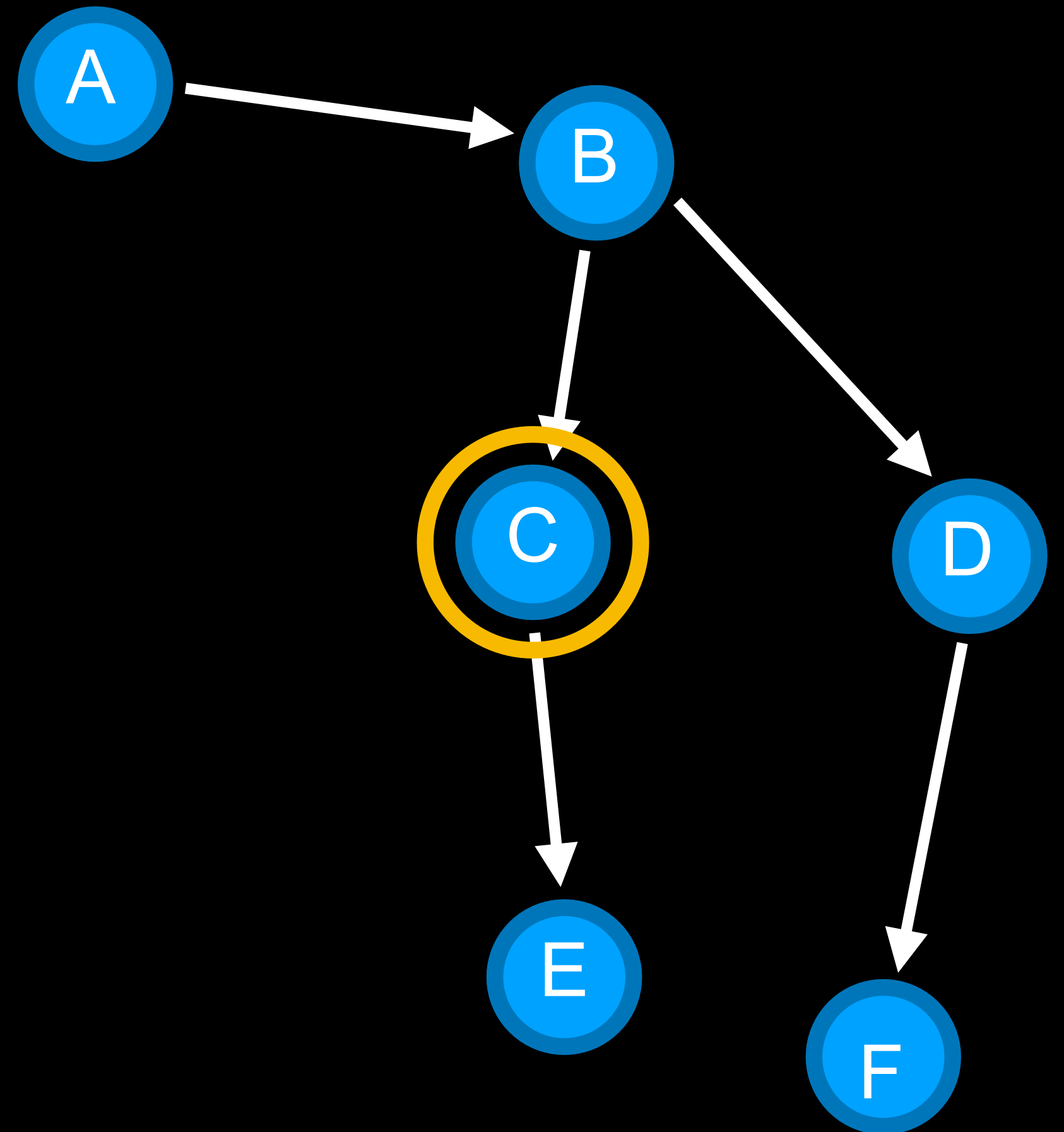
Conjunto explorado



Camino de A a E?

Frontera

Conjunto explorado

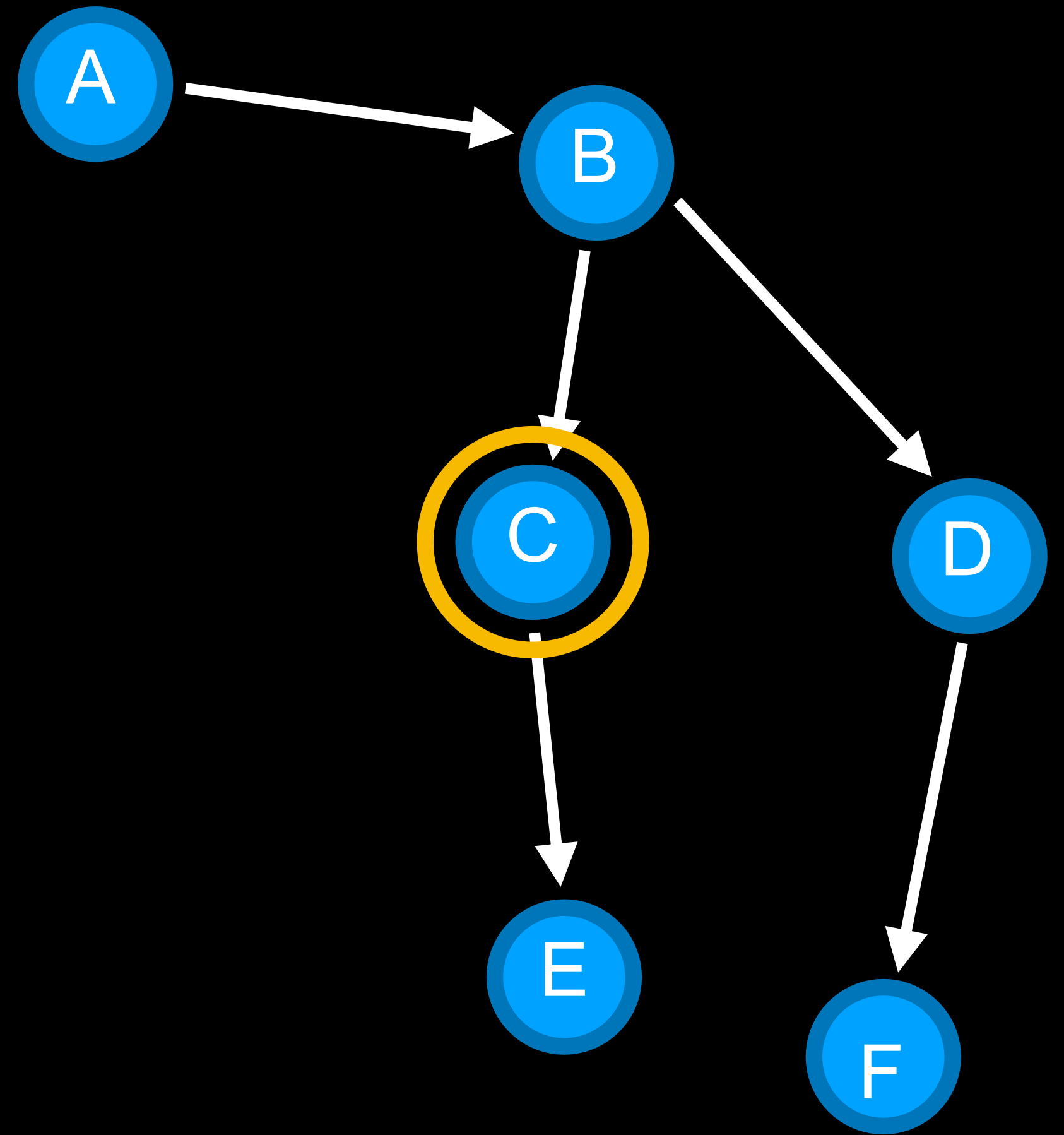


Camino de A a E?

Frontera



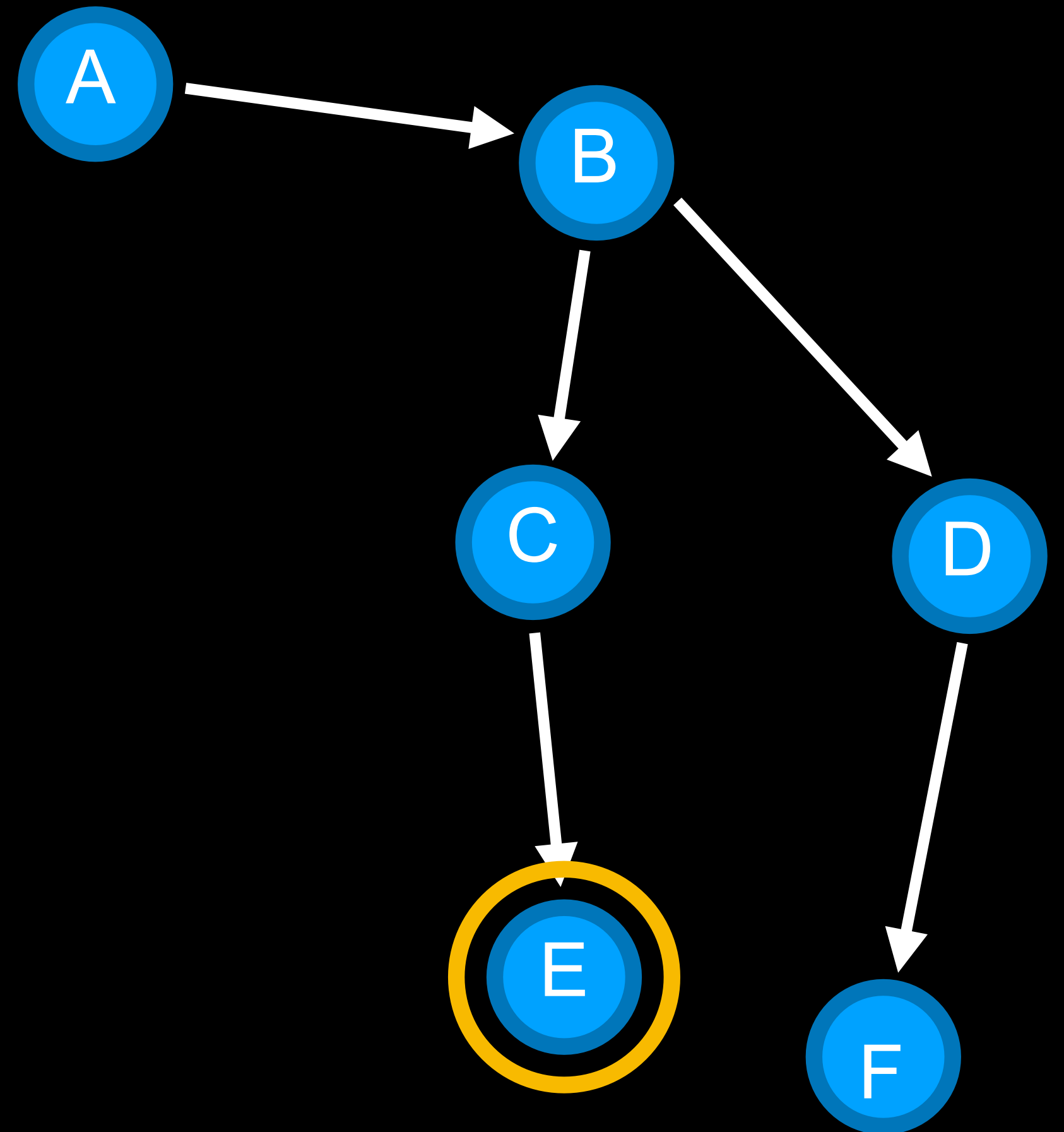
Conjunto explorado



Camino de A a E?

Frontera

Conjunto explorado



Búsqueda en profundidad

Depth-First Search

línea 127 StackFrontier LIFO

Búsqueda en profundidad

algoritmo de búsqueda que siempre expande
el nodo más profundo en la frontera

Búsqueda en amplitud

Breadth-First Search

línea 127 QueueFrontier FIFO

Búsqueda en amplitud

algoritmo de búsqueda que siempre expande
el nodo más superficial en la frontera

Cola

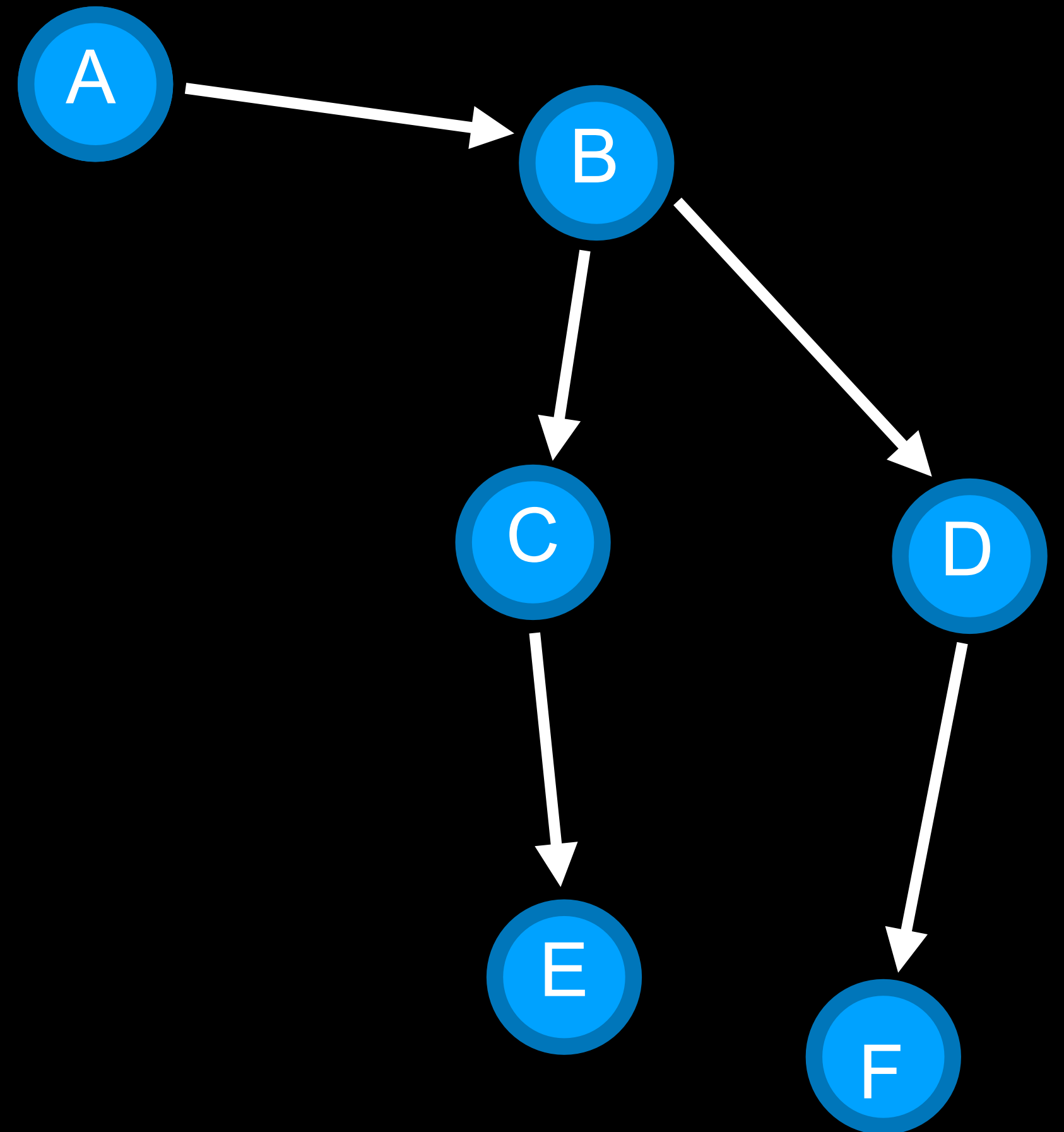
Tipo de dato: Primero en entrar, primero en salir

first-in first-out data type

Find a path from A to E.

Frontier

Explored Set

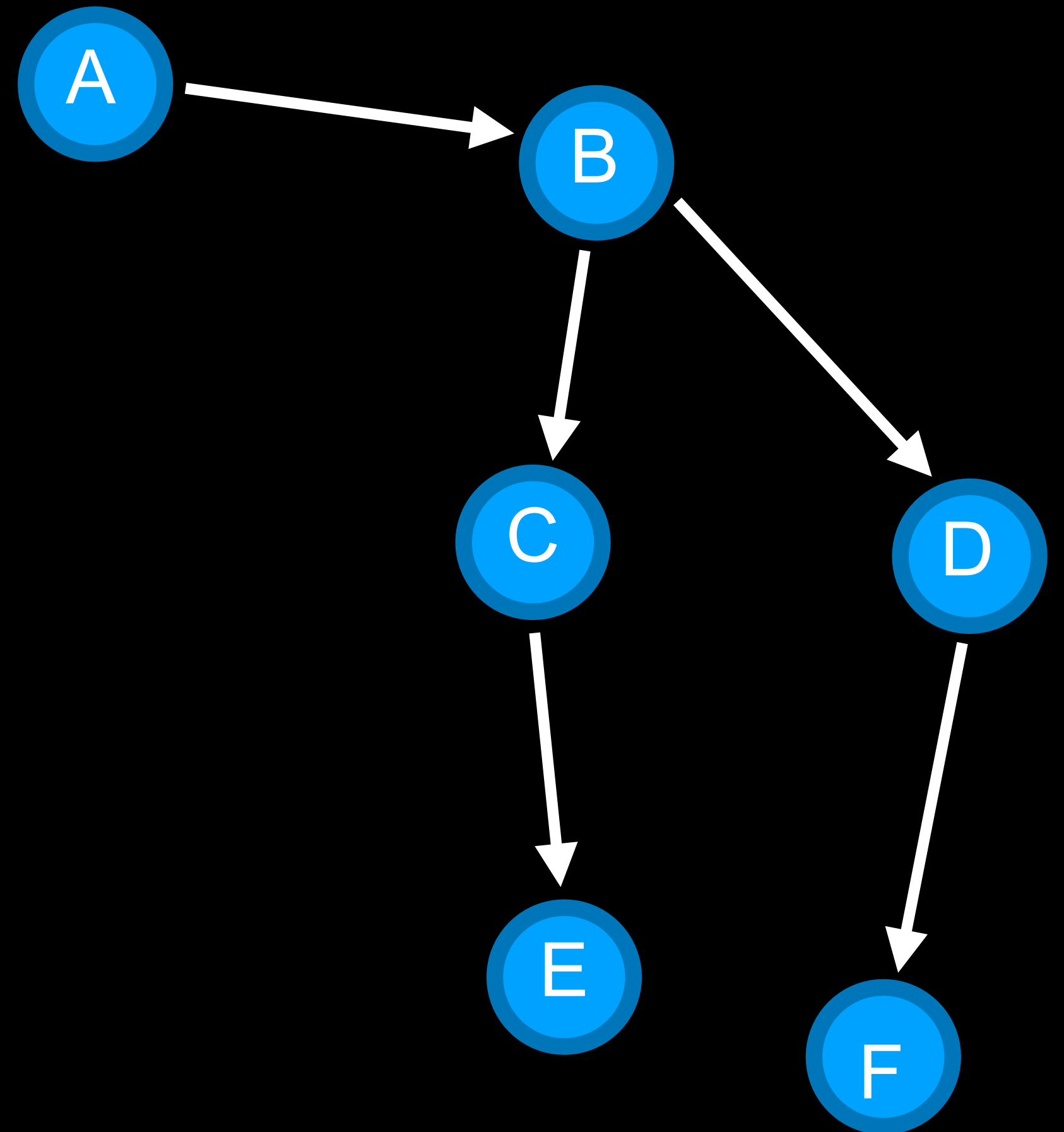


Find a path from A to E.

Frontier



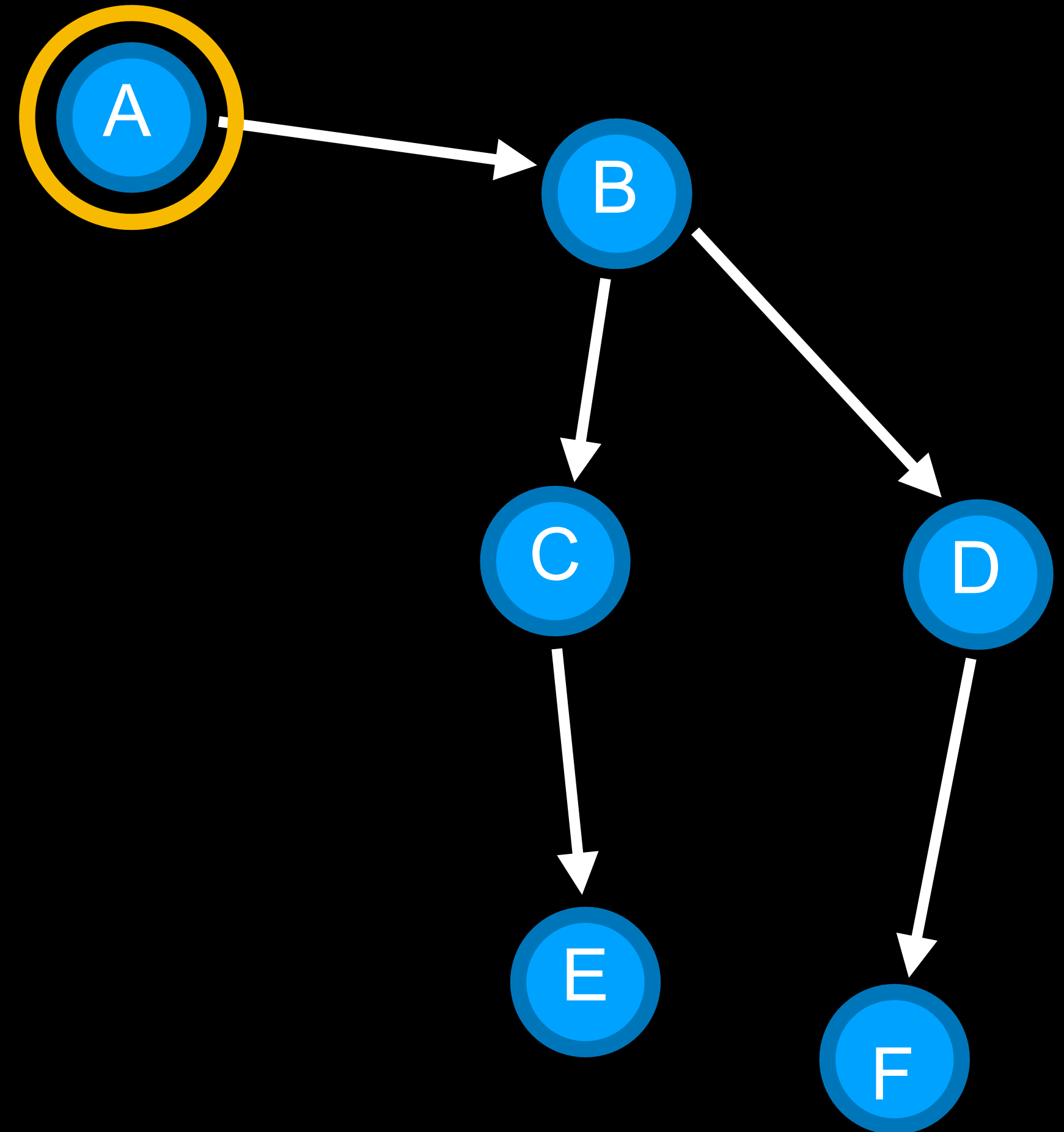
Explored Set



Find a path from A to E.

Frontier

Explored Set

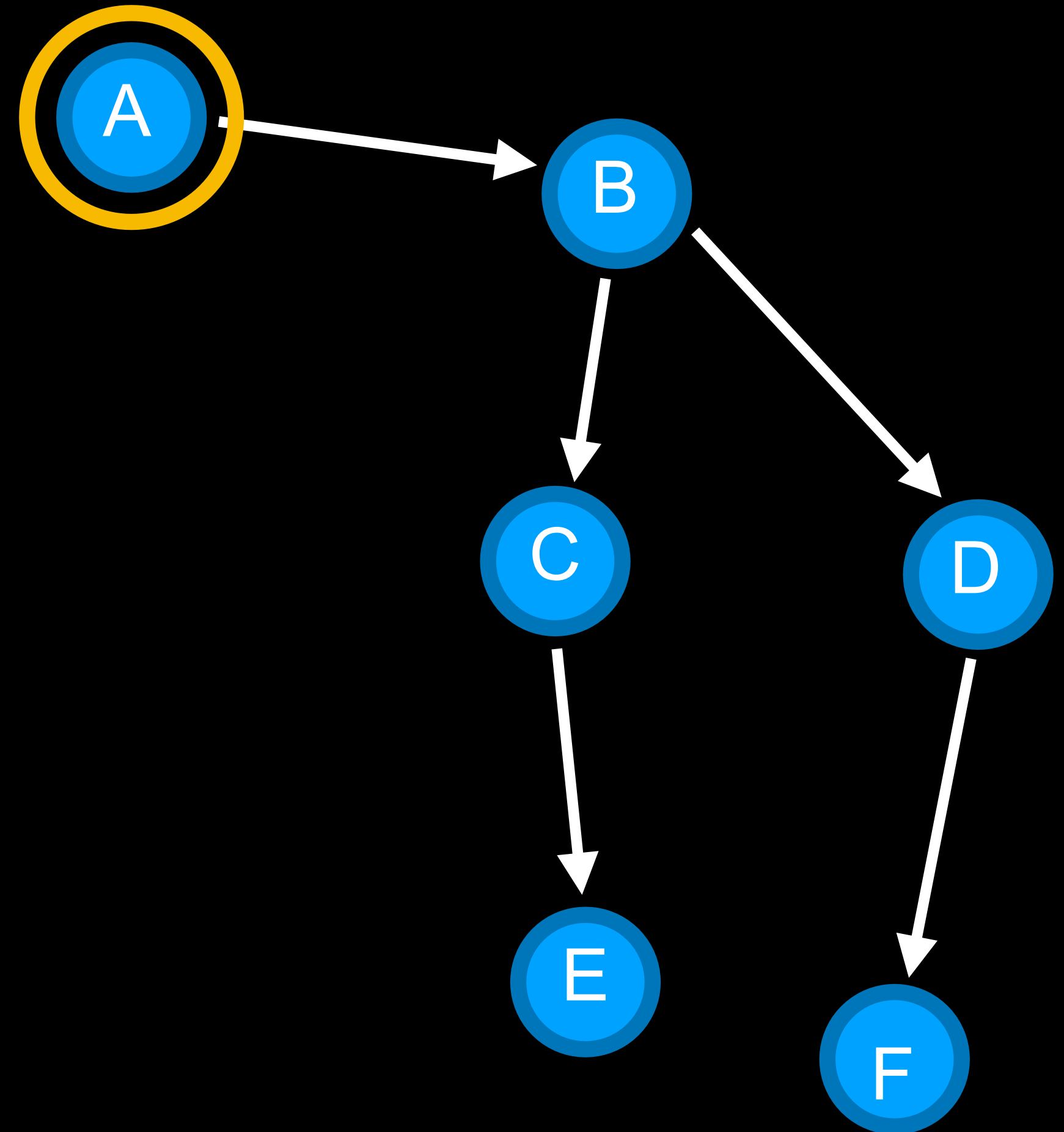


Find a path from A to E.

Frontier



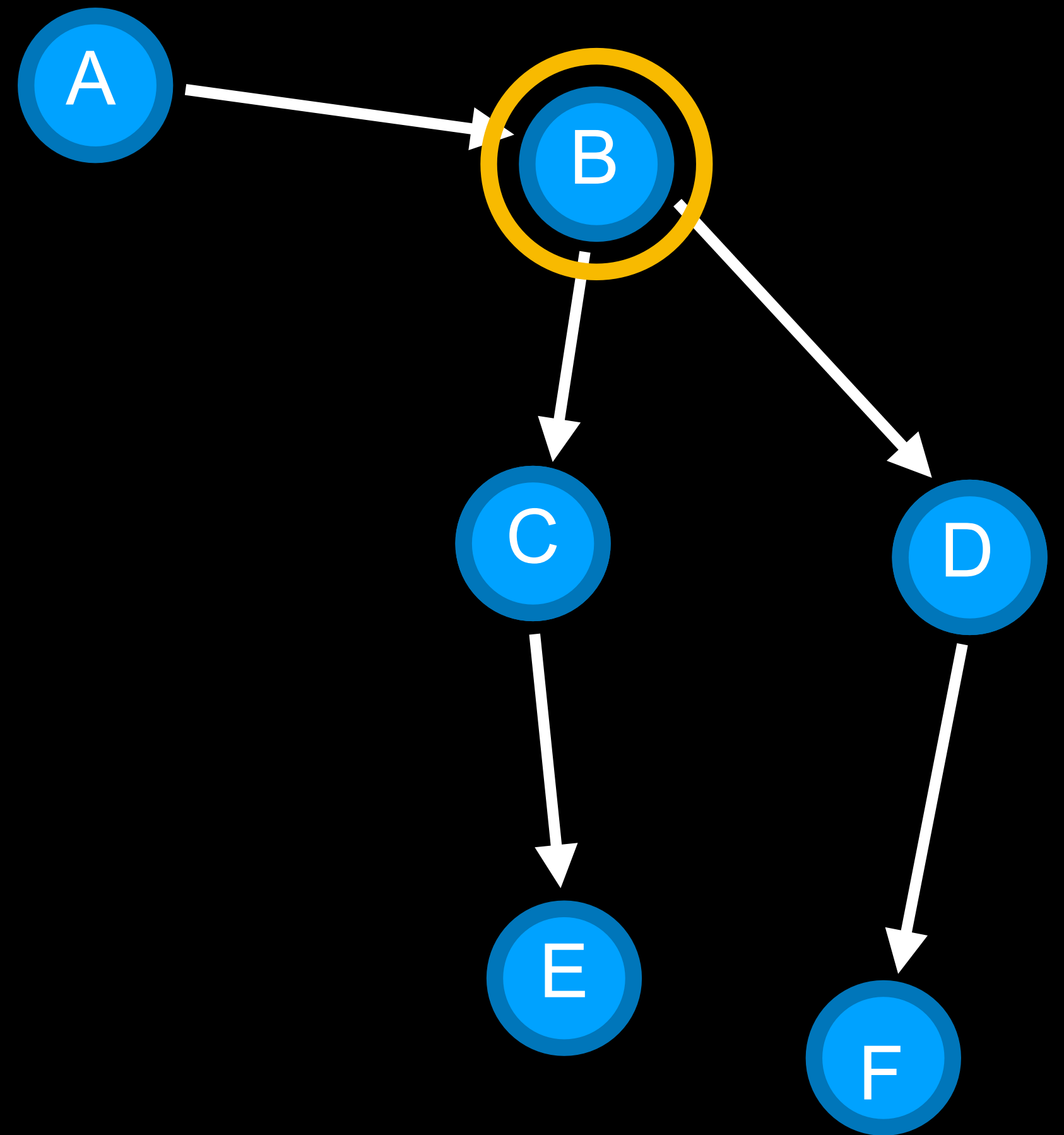
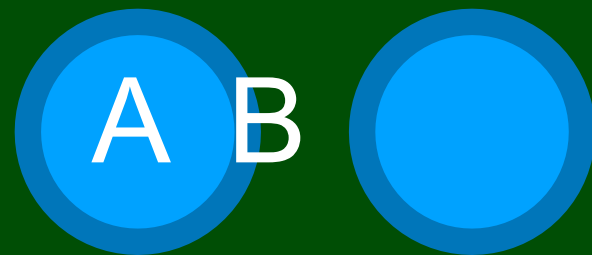
Explored Set



Find a path from A to E.

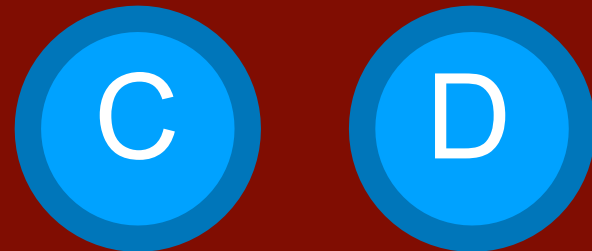
Frontier

Explored Set

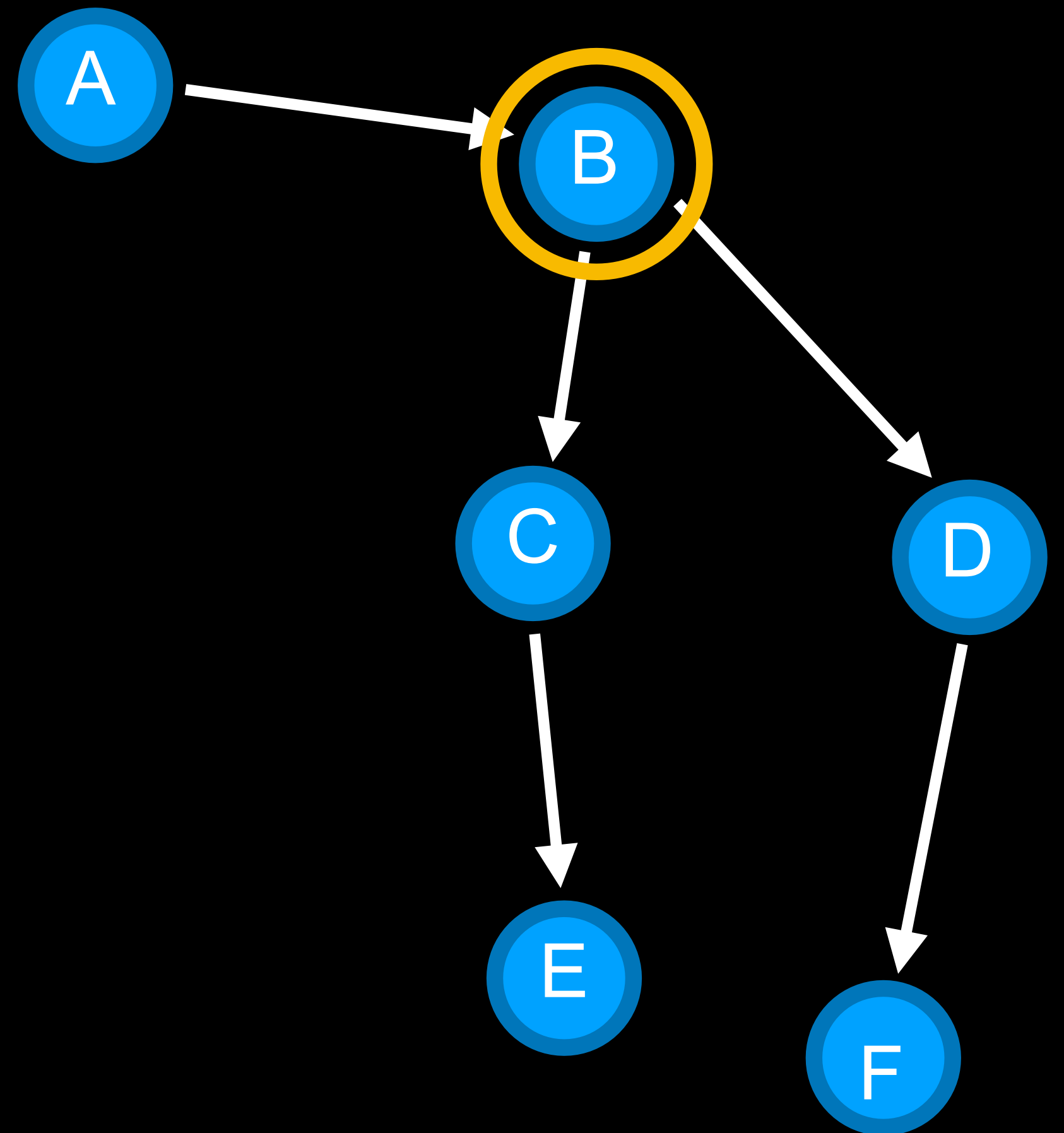
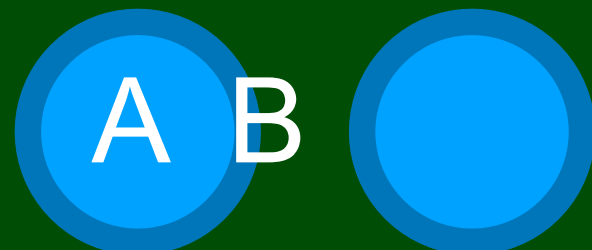


Find a path from A to E.

Frontier



Explored Set

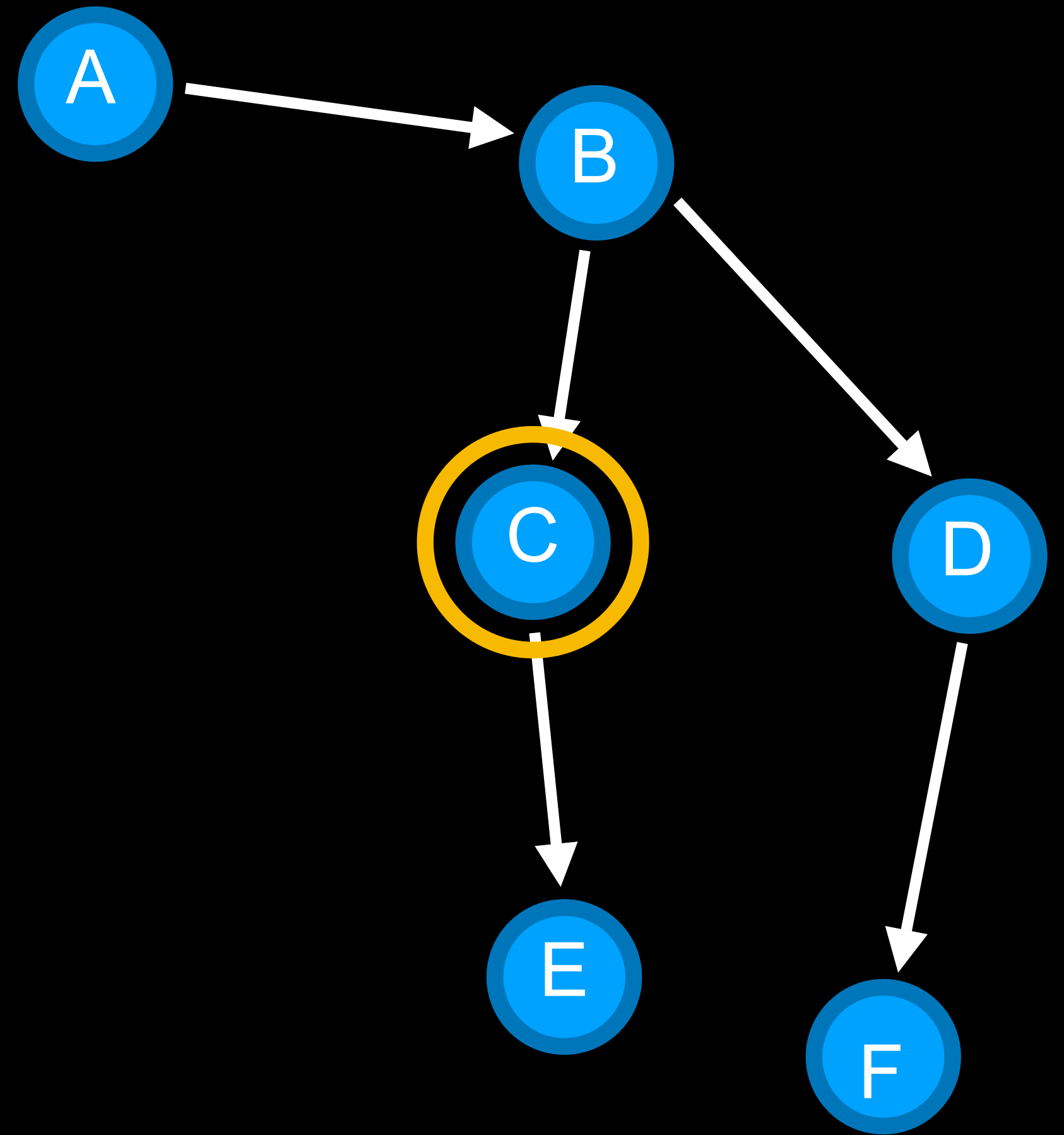
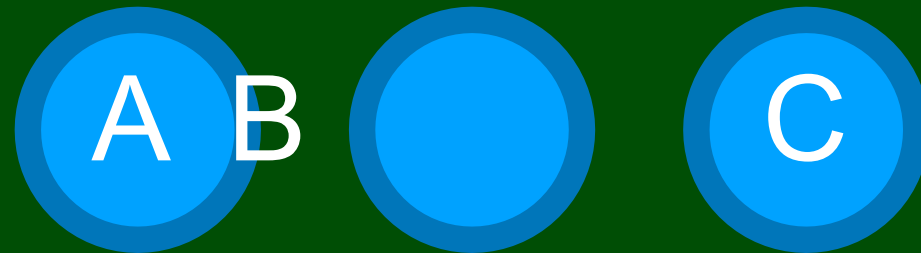


Find a path from A to E.

Frontier

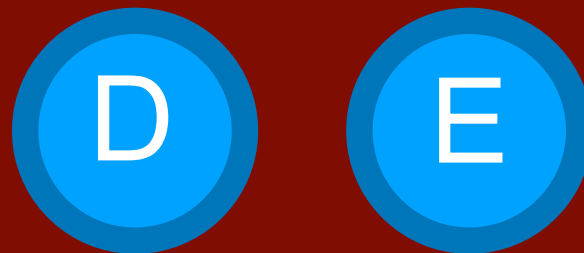


Explored Set

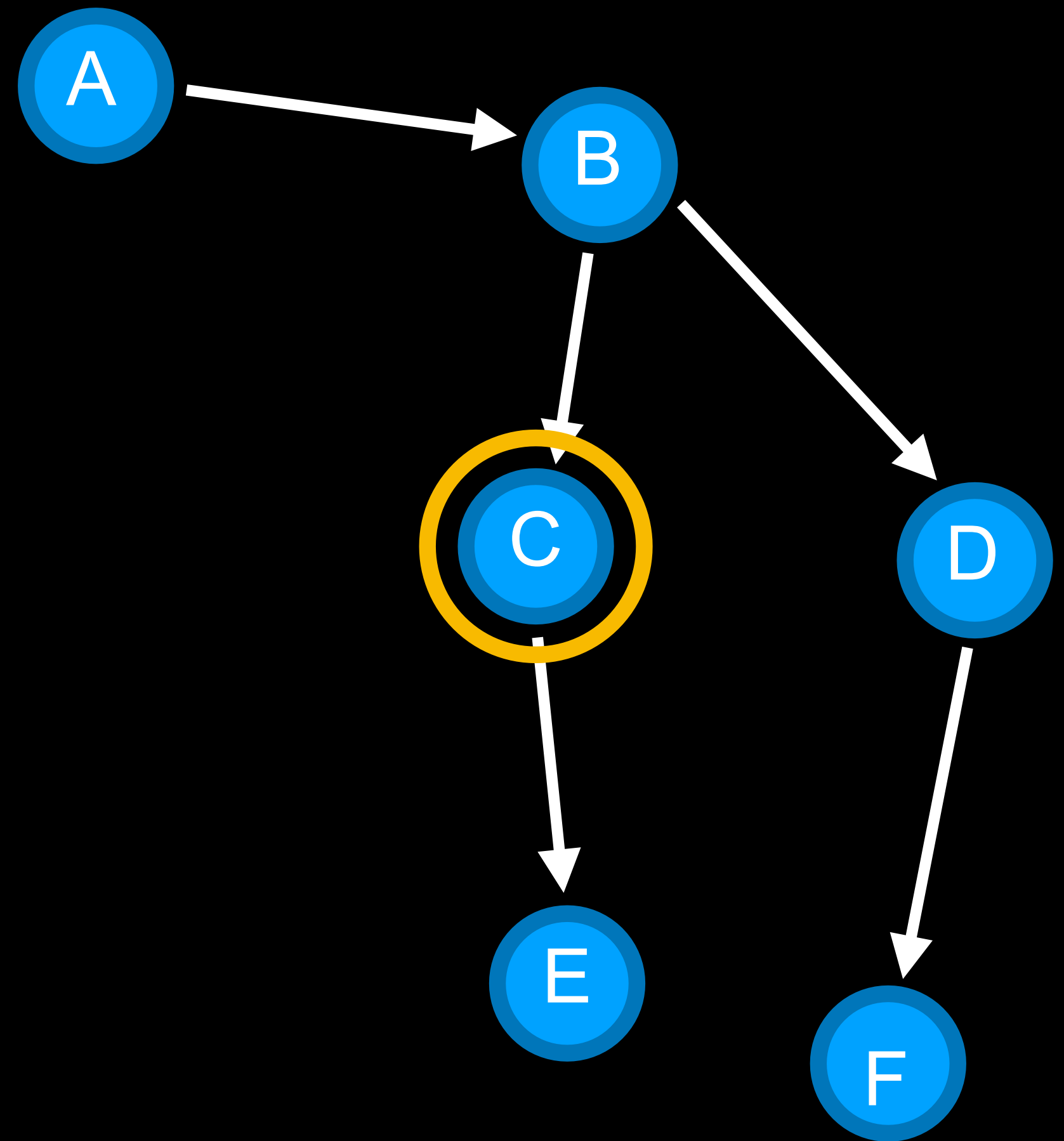
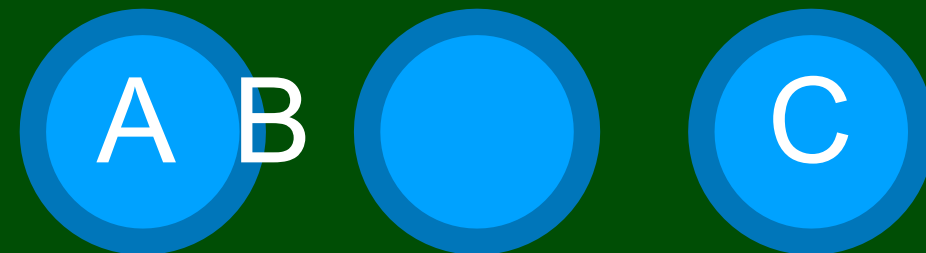


Find a path from A to E.

Frontier



Explored Set

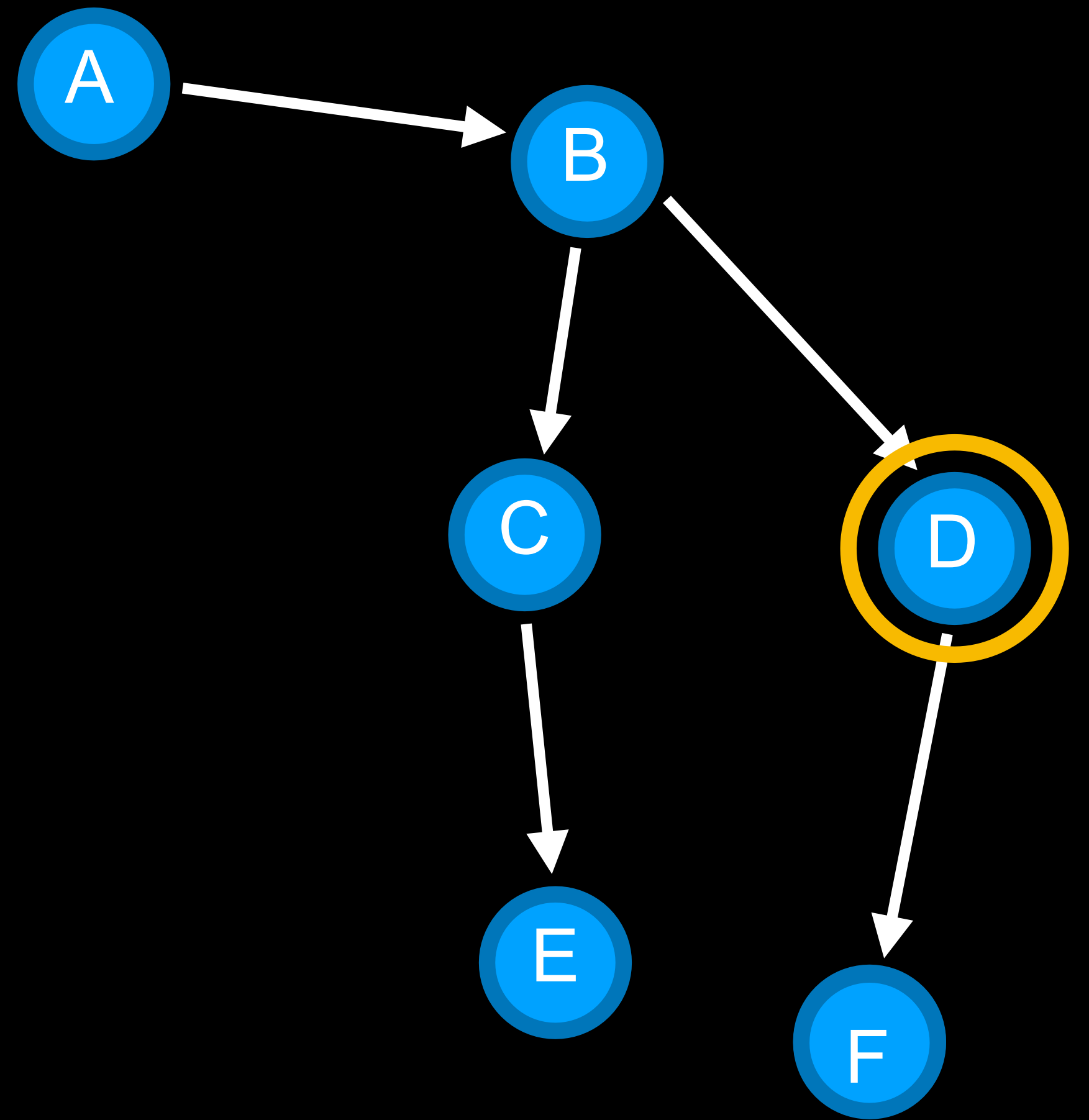
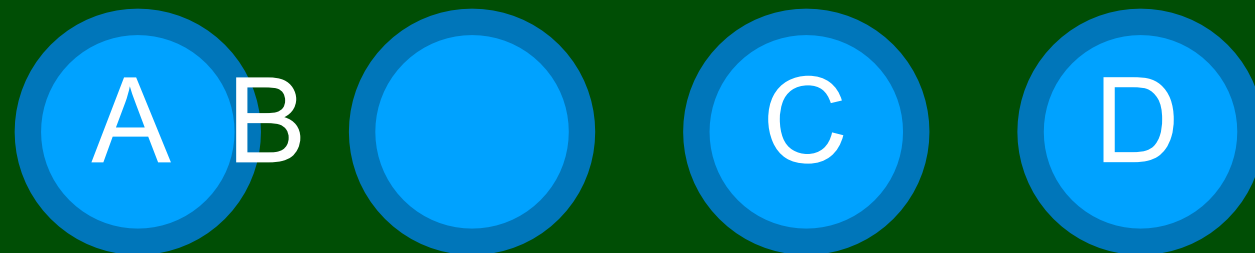


Find a path from A to E.

Frontier

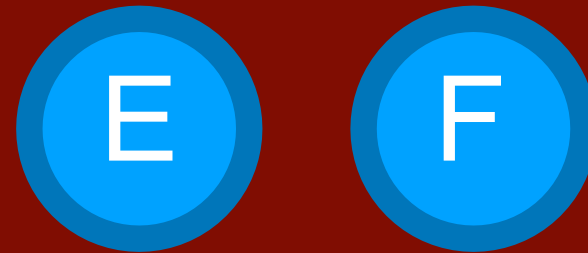


Explored Set

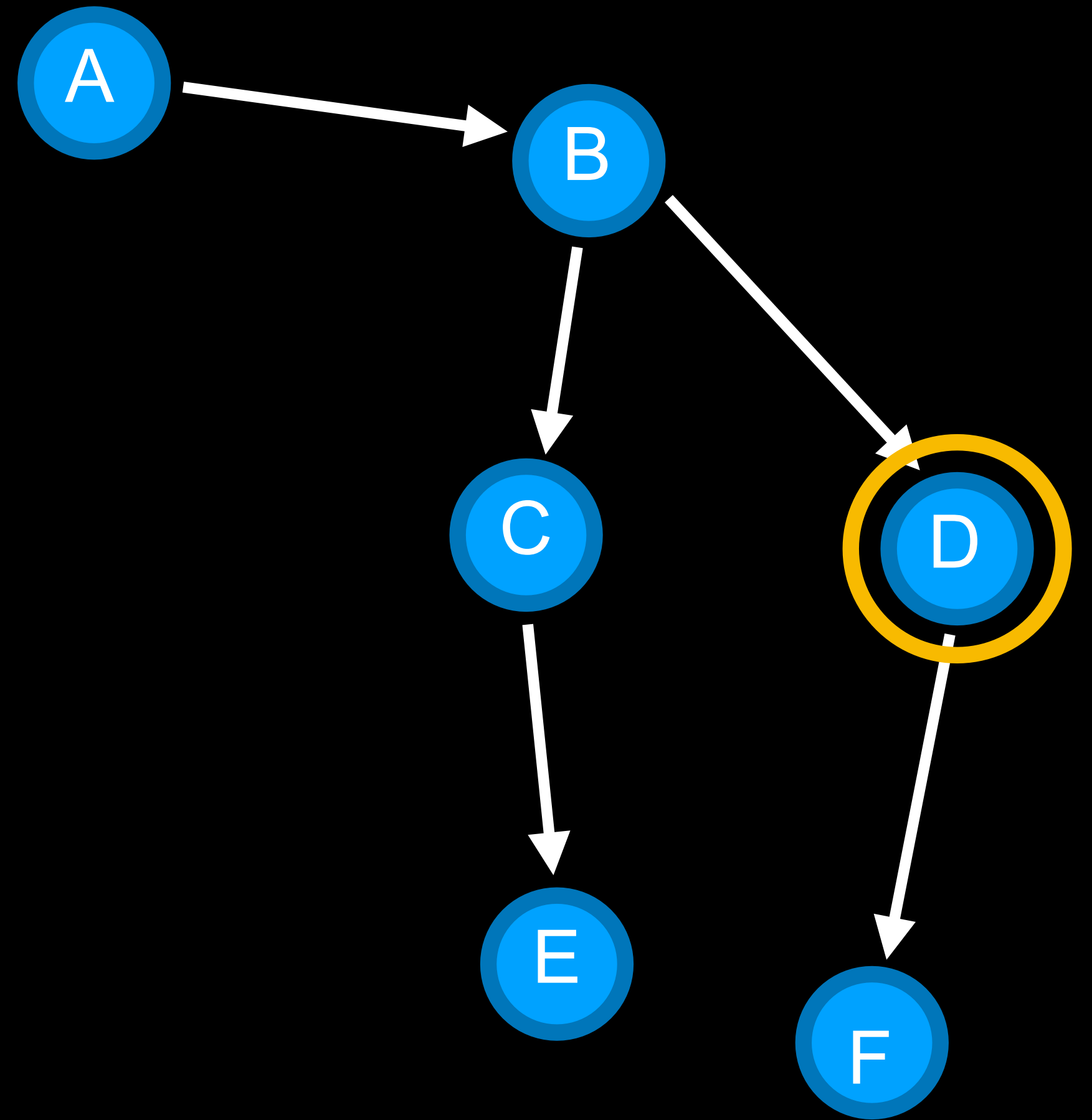
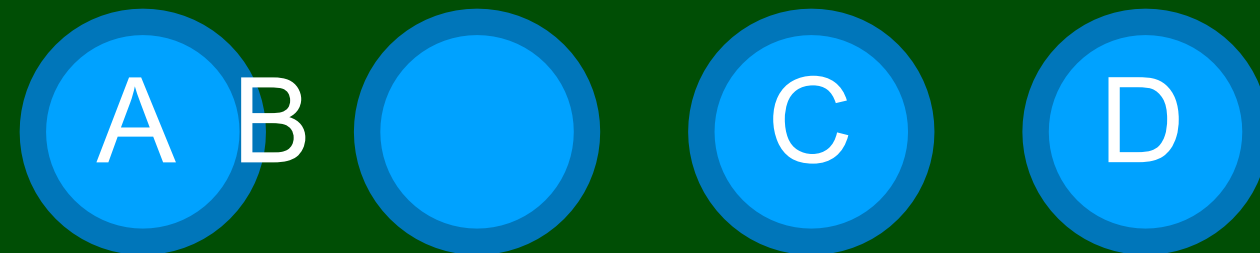


Find a path from A to E.

Frontier



Explored Set

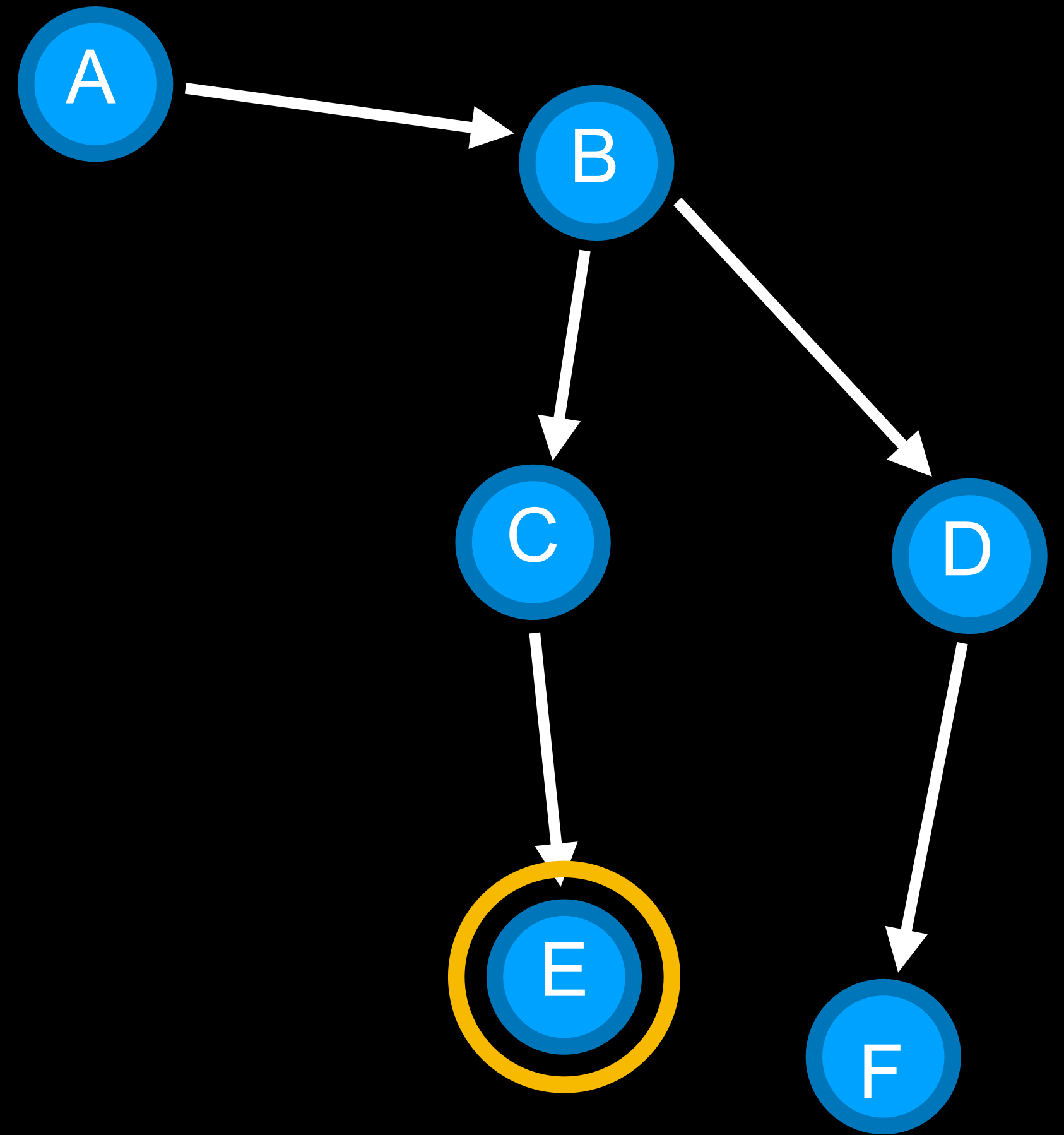
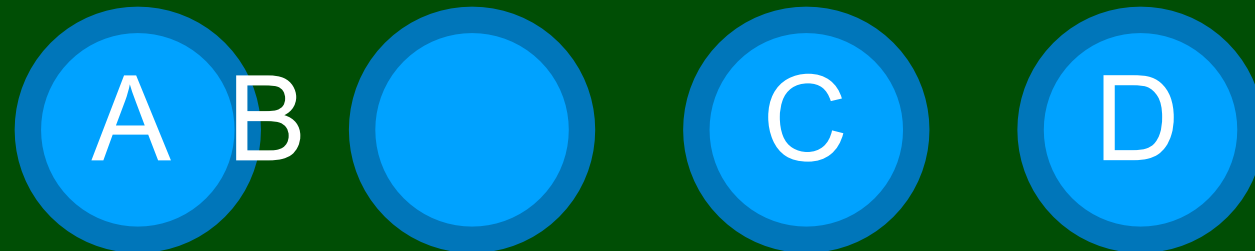


Find a path from A to E.

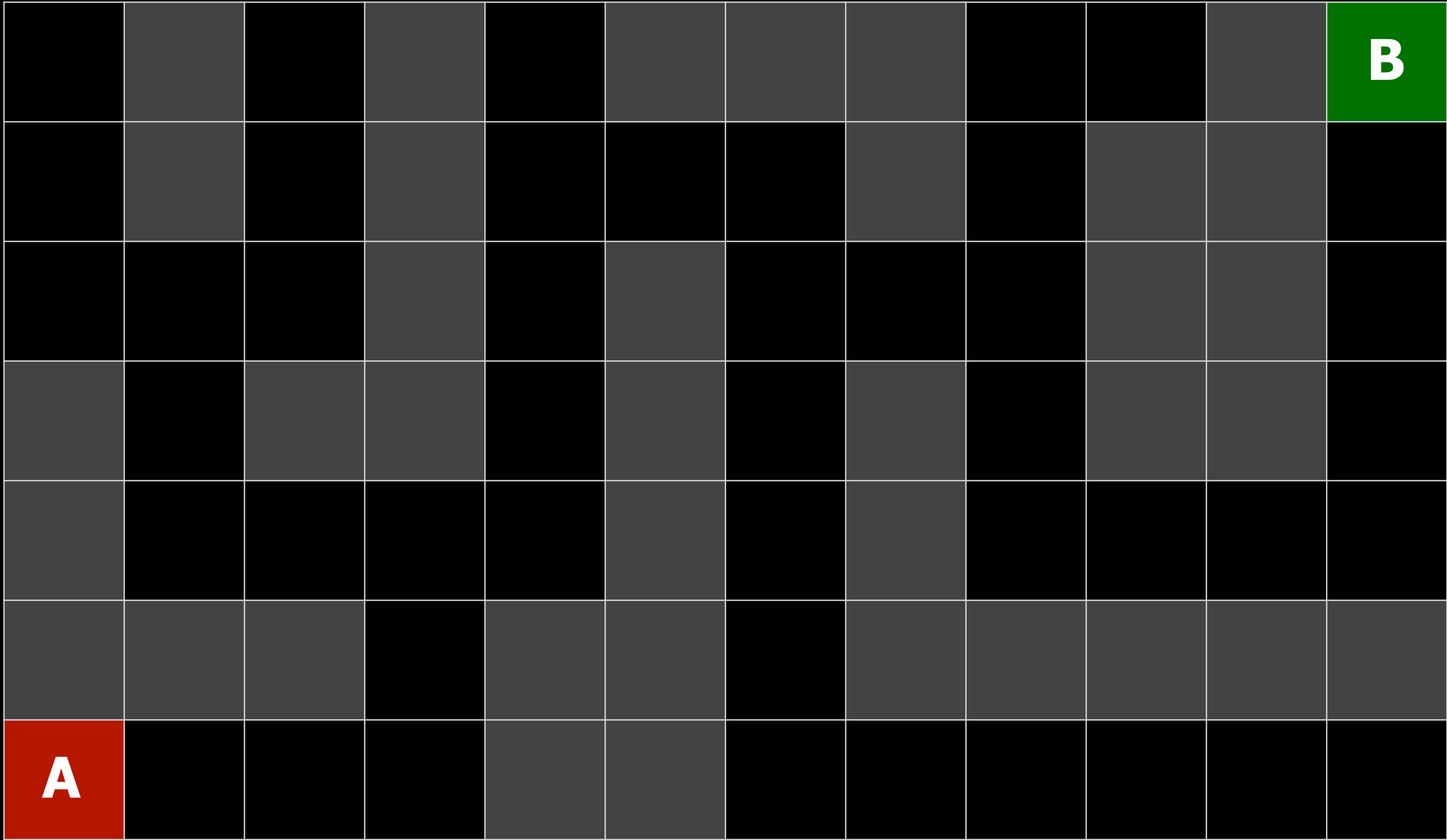
Frontier



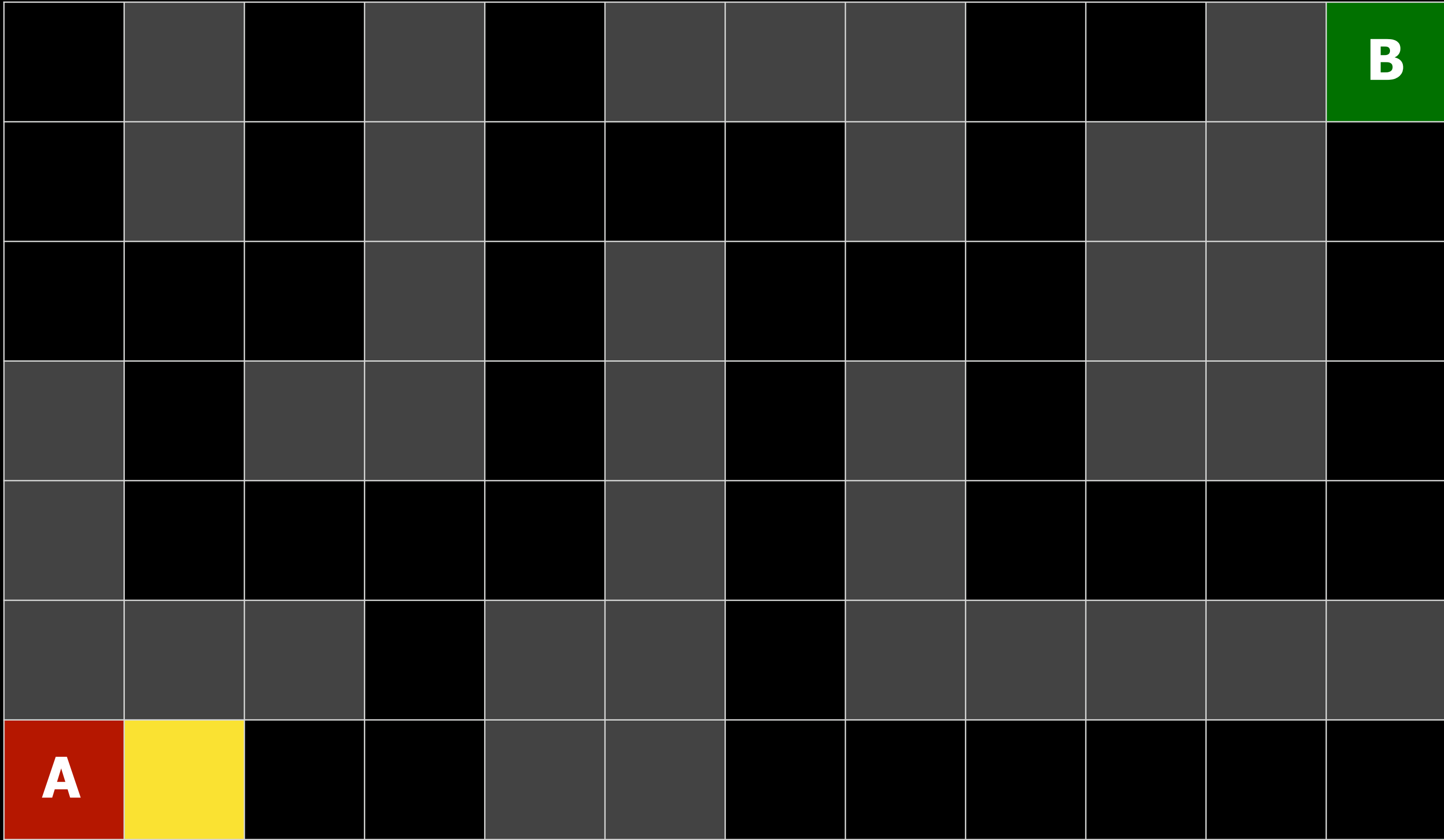
Explored Set



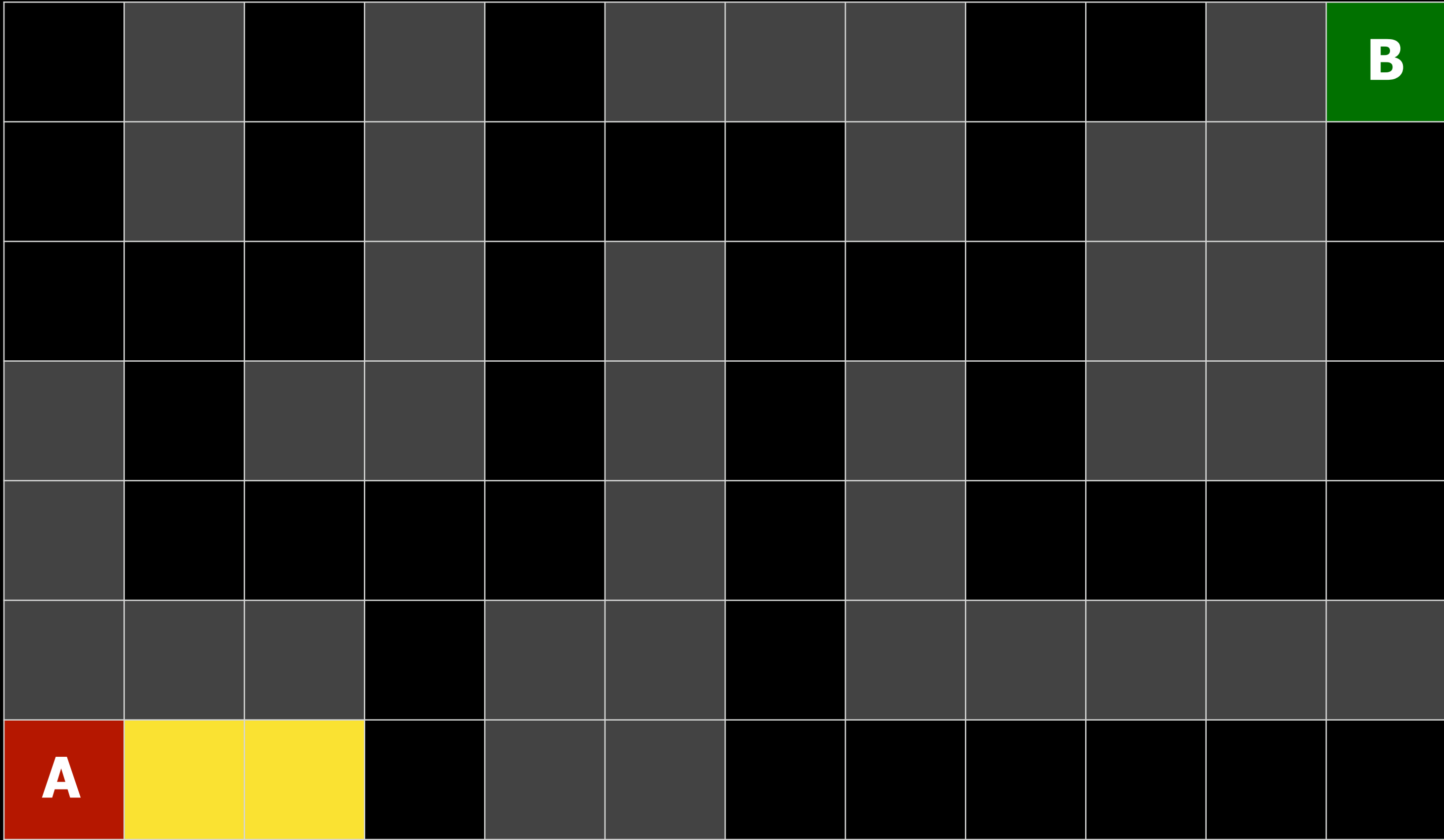
Depth-First Search



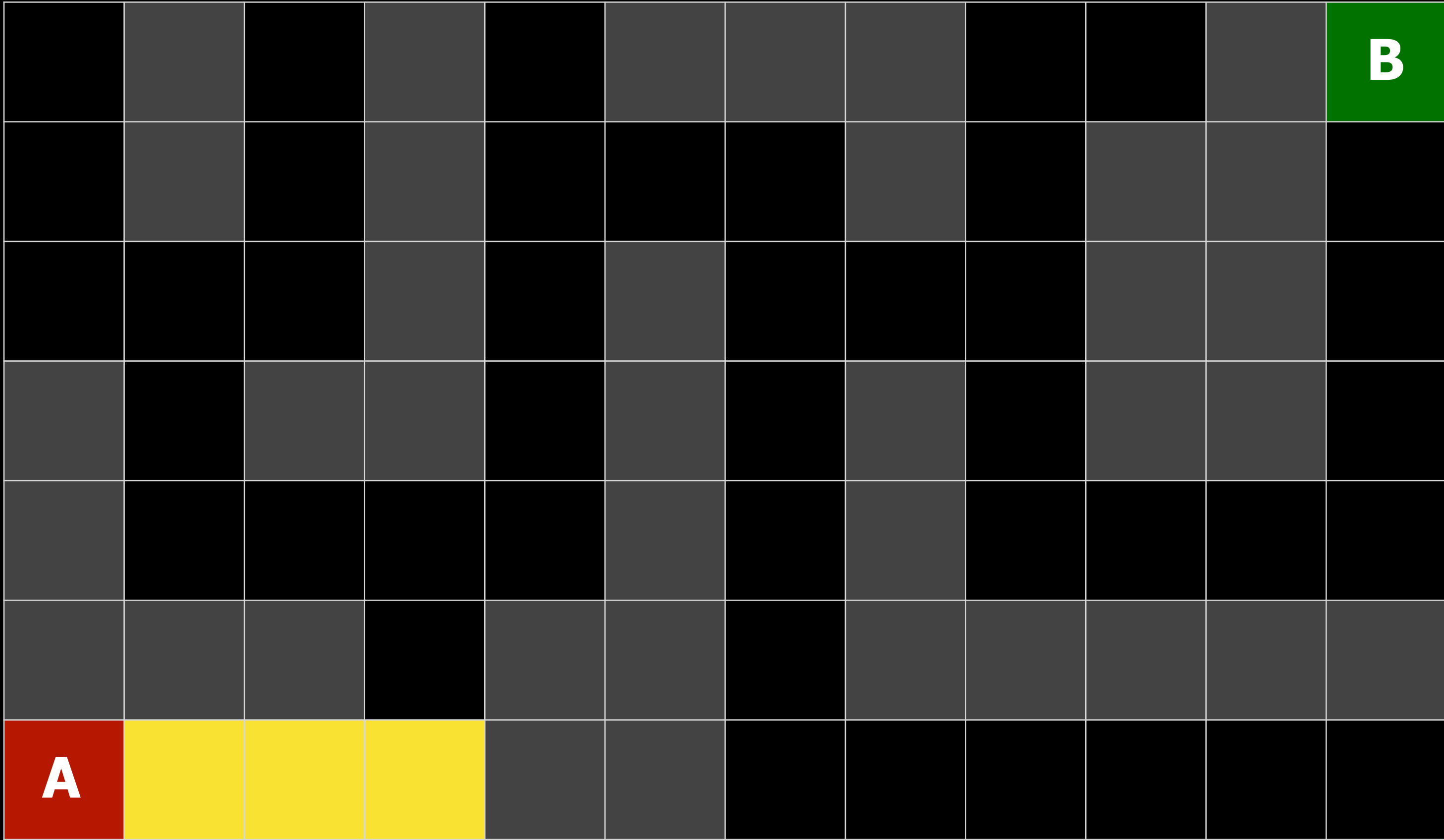
Depth-First Search



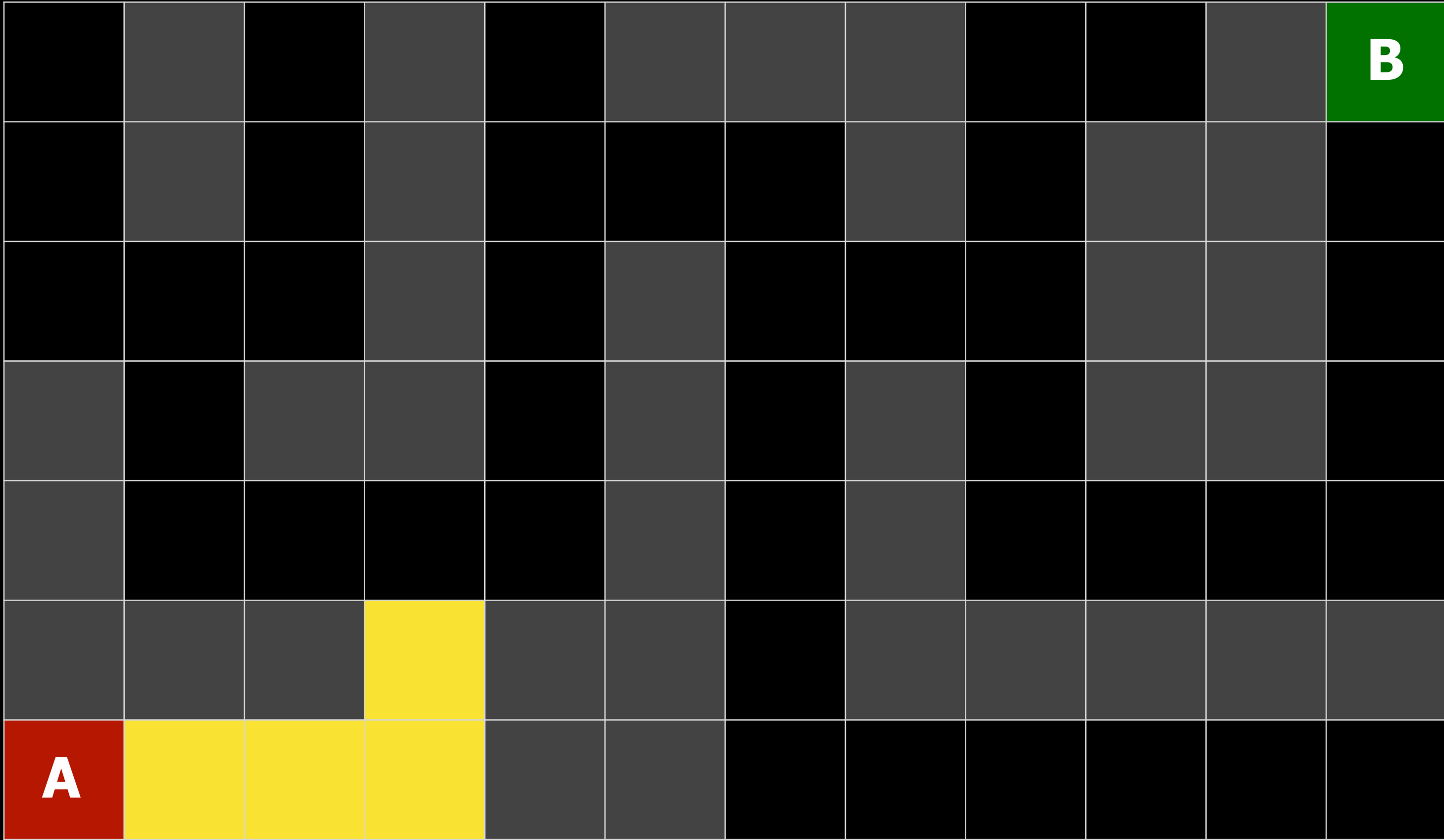
Depth-First Search



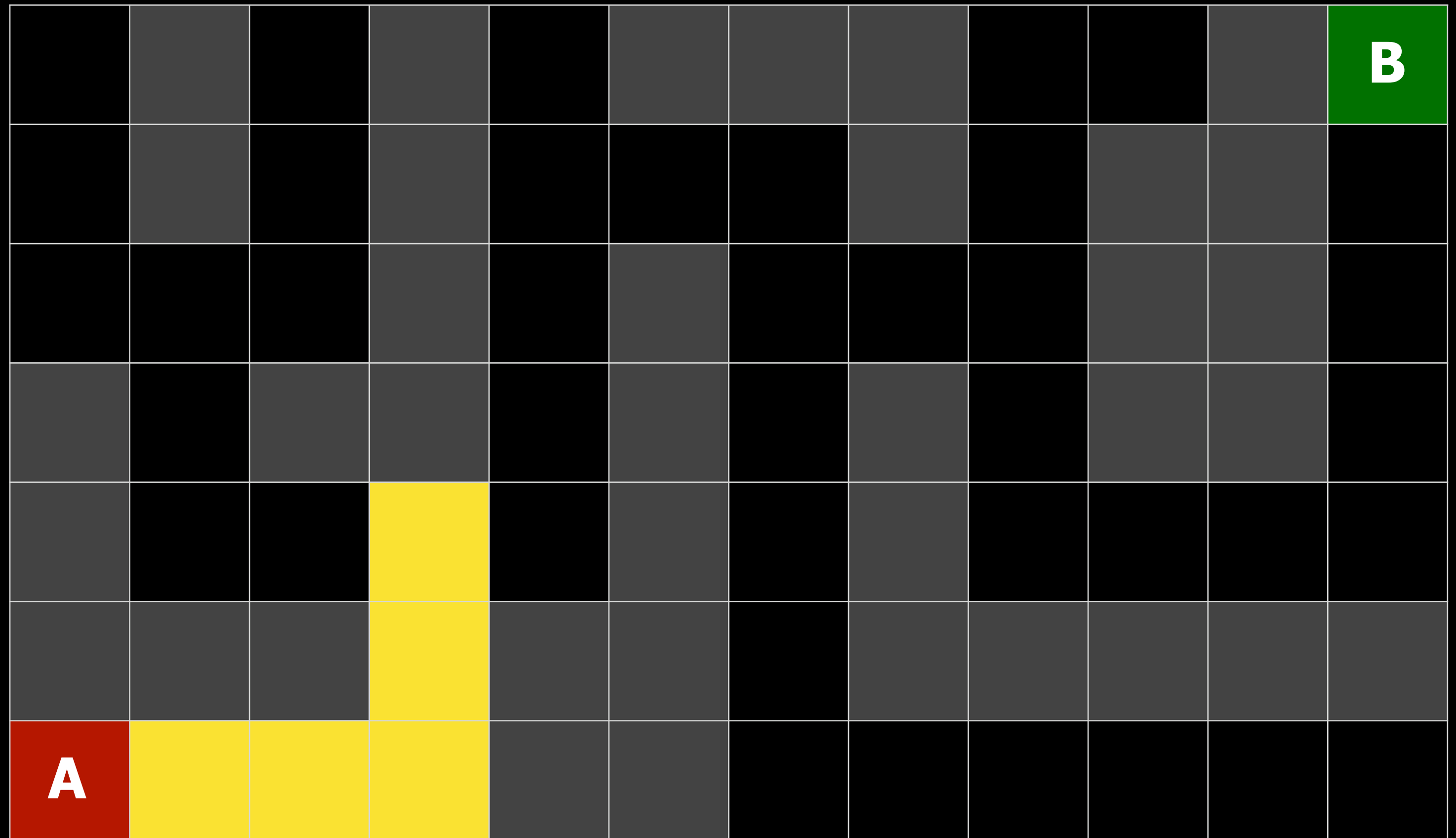
Depth-First Search



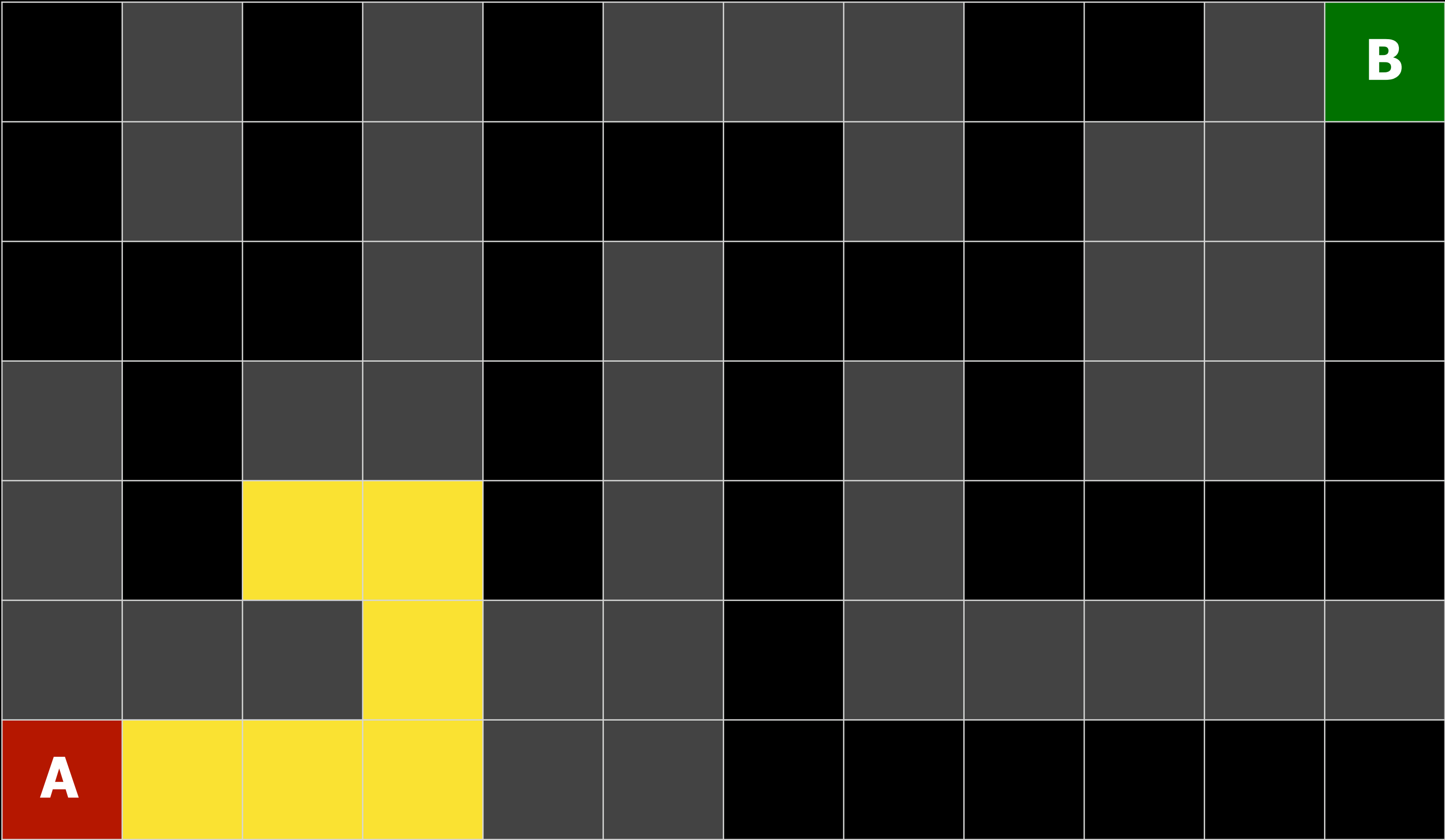
Depth-First Search



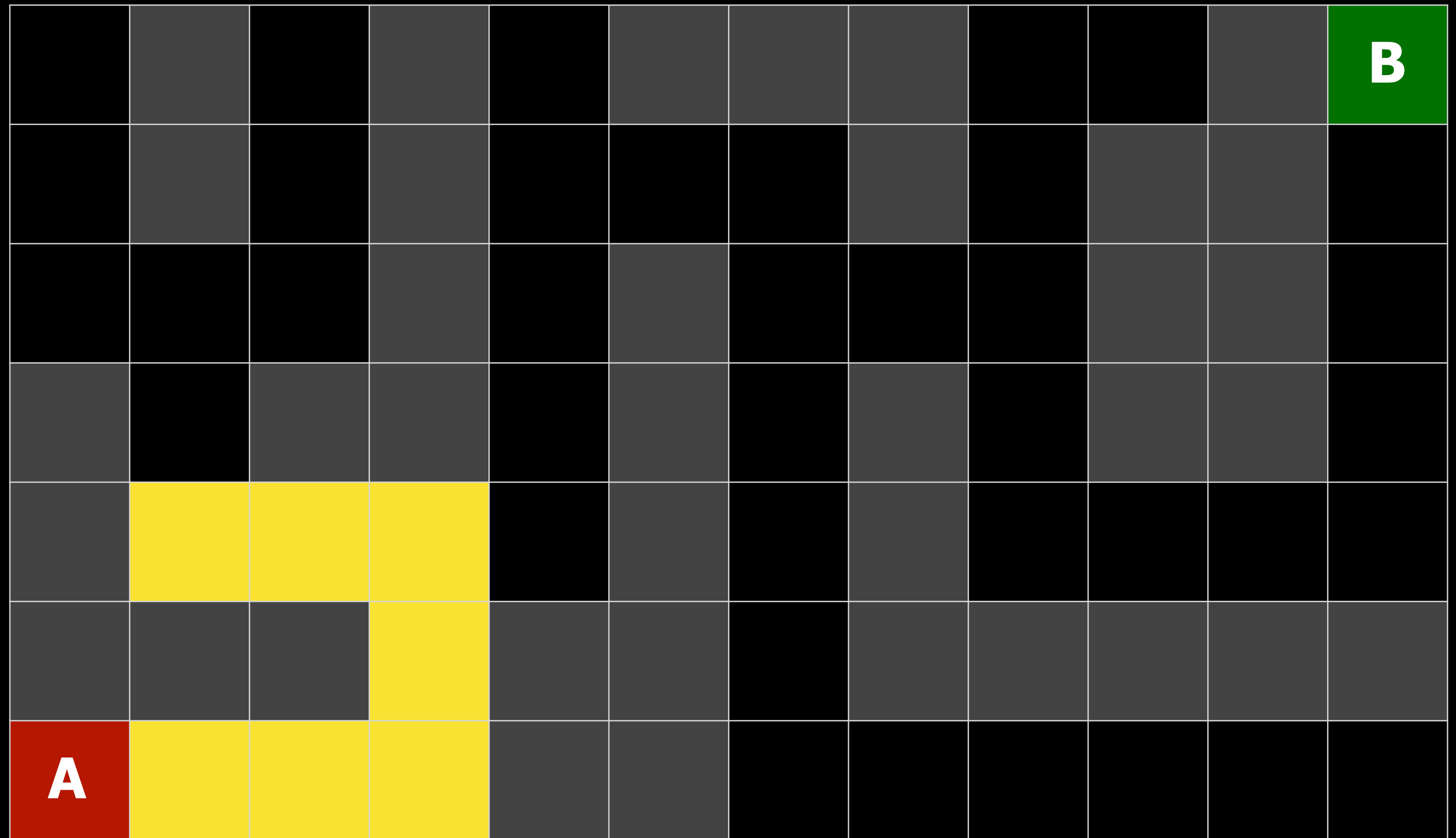
Depth-First Search



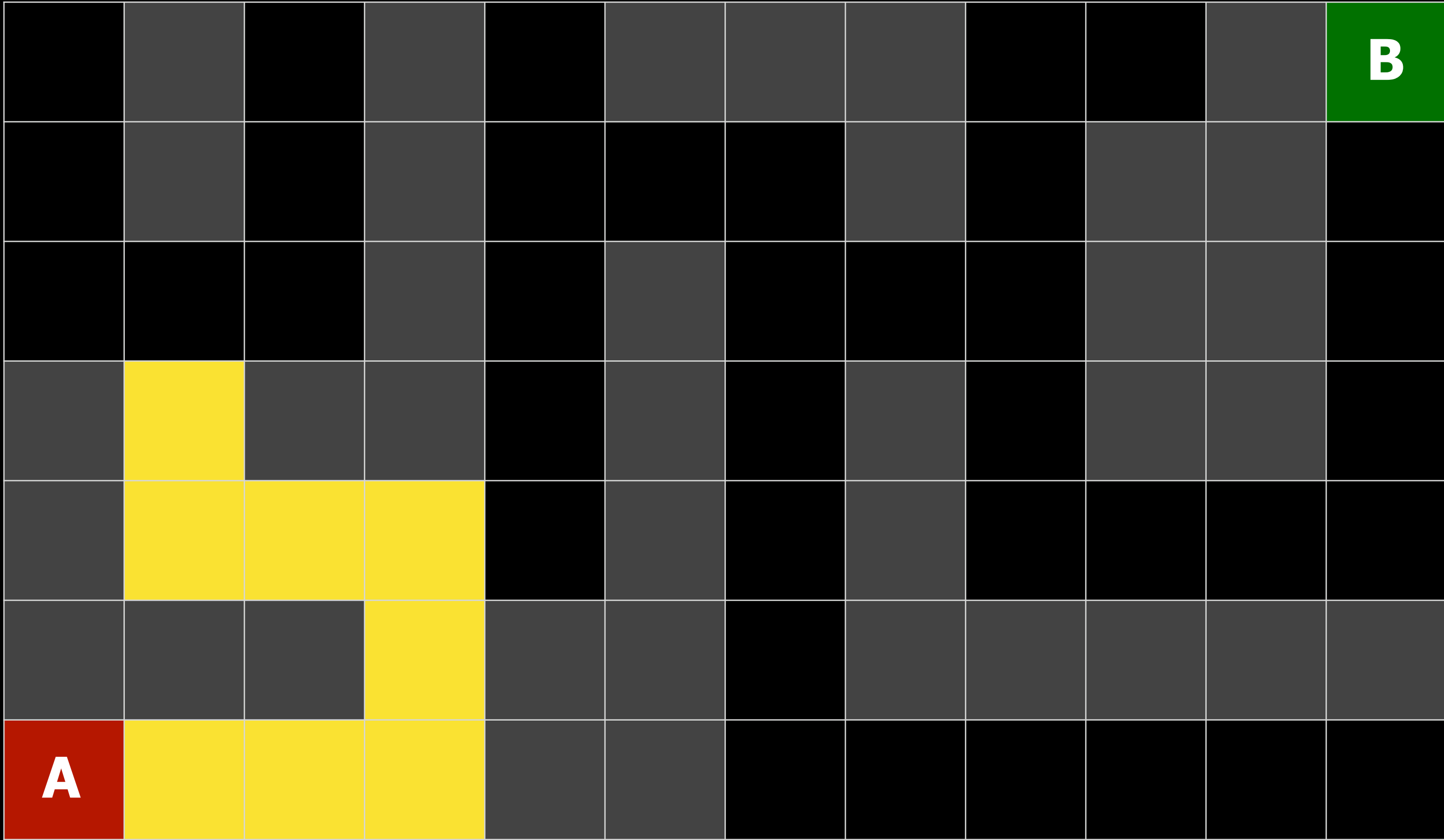
Depth-First Search



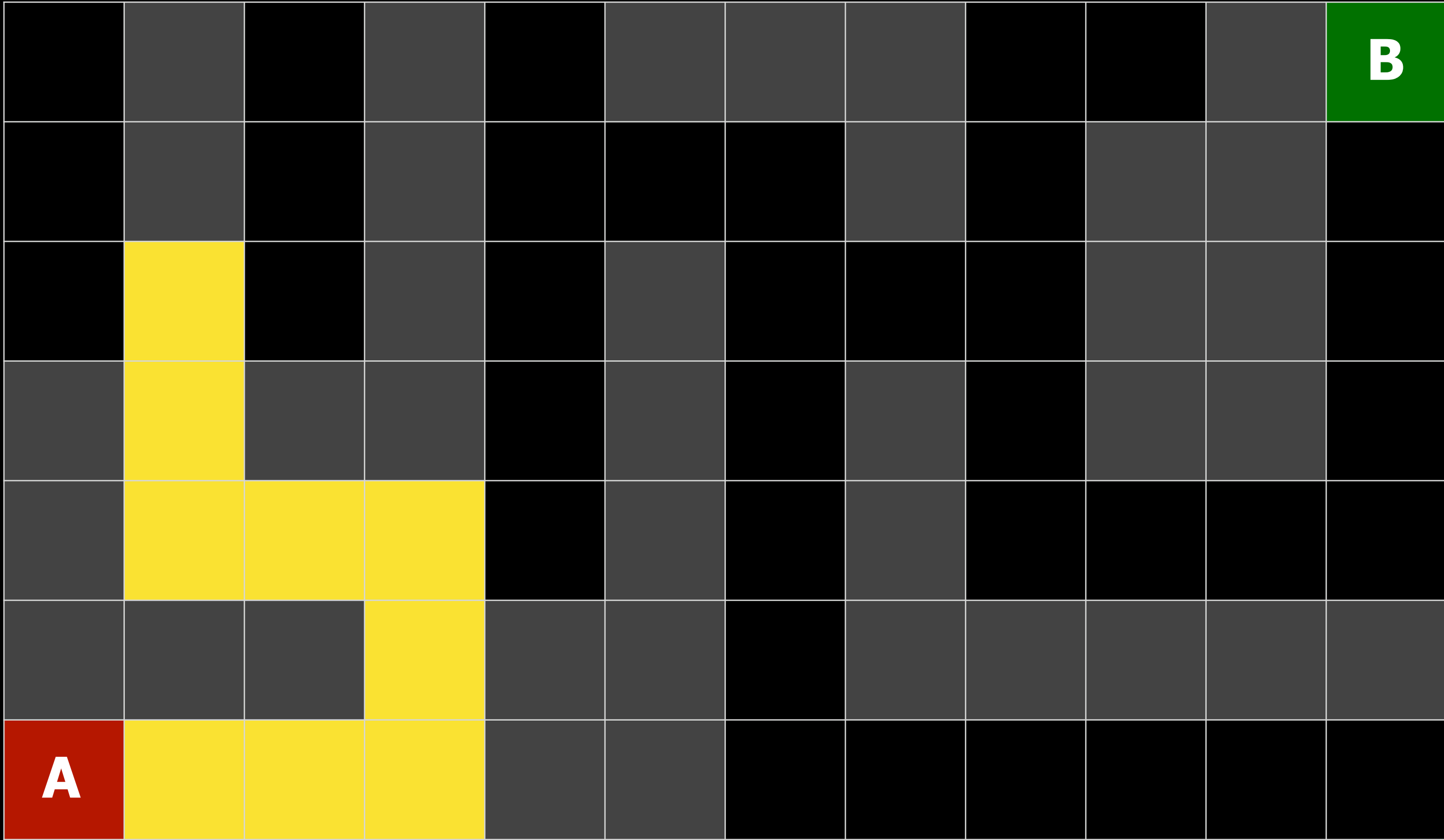
Depth-First Search



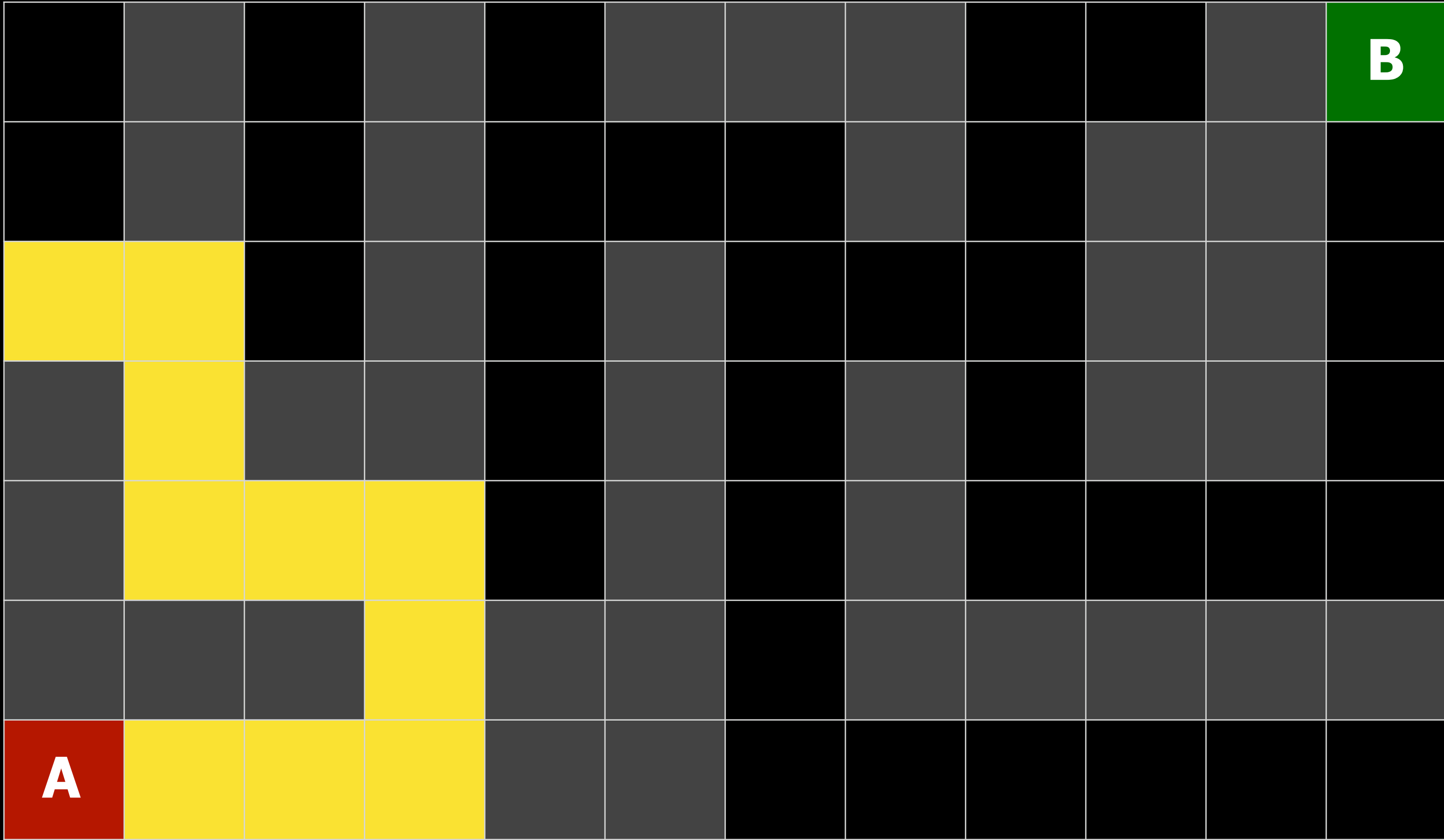
Depth-First Search



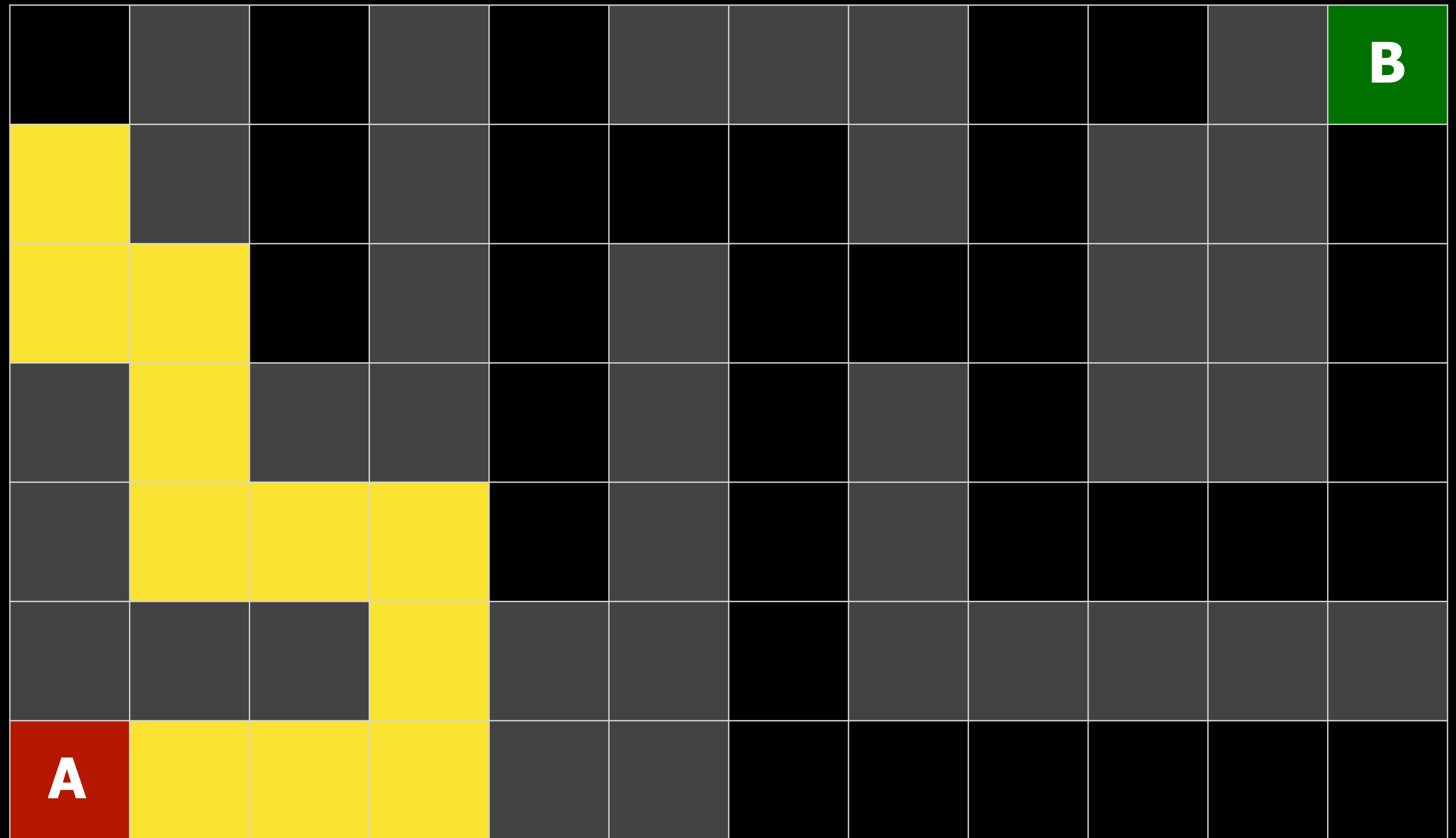
Depth-First Search



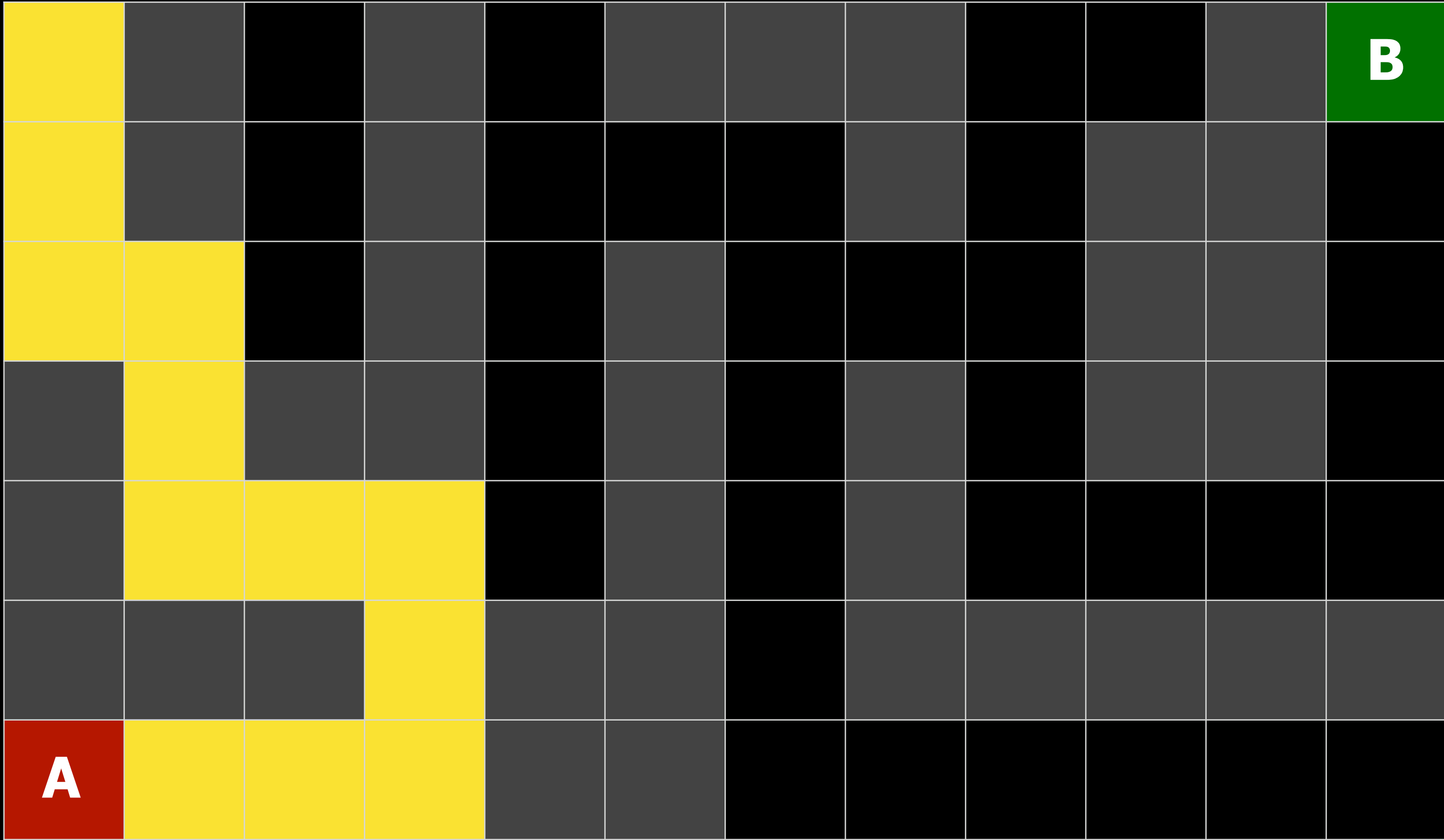
Depth-First Search



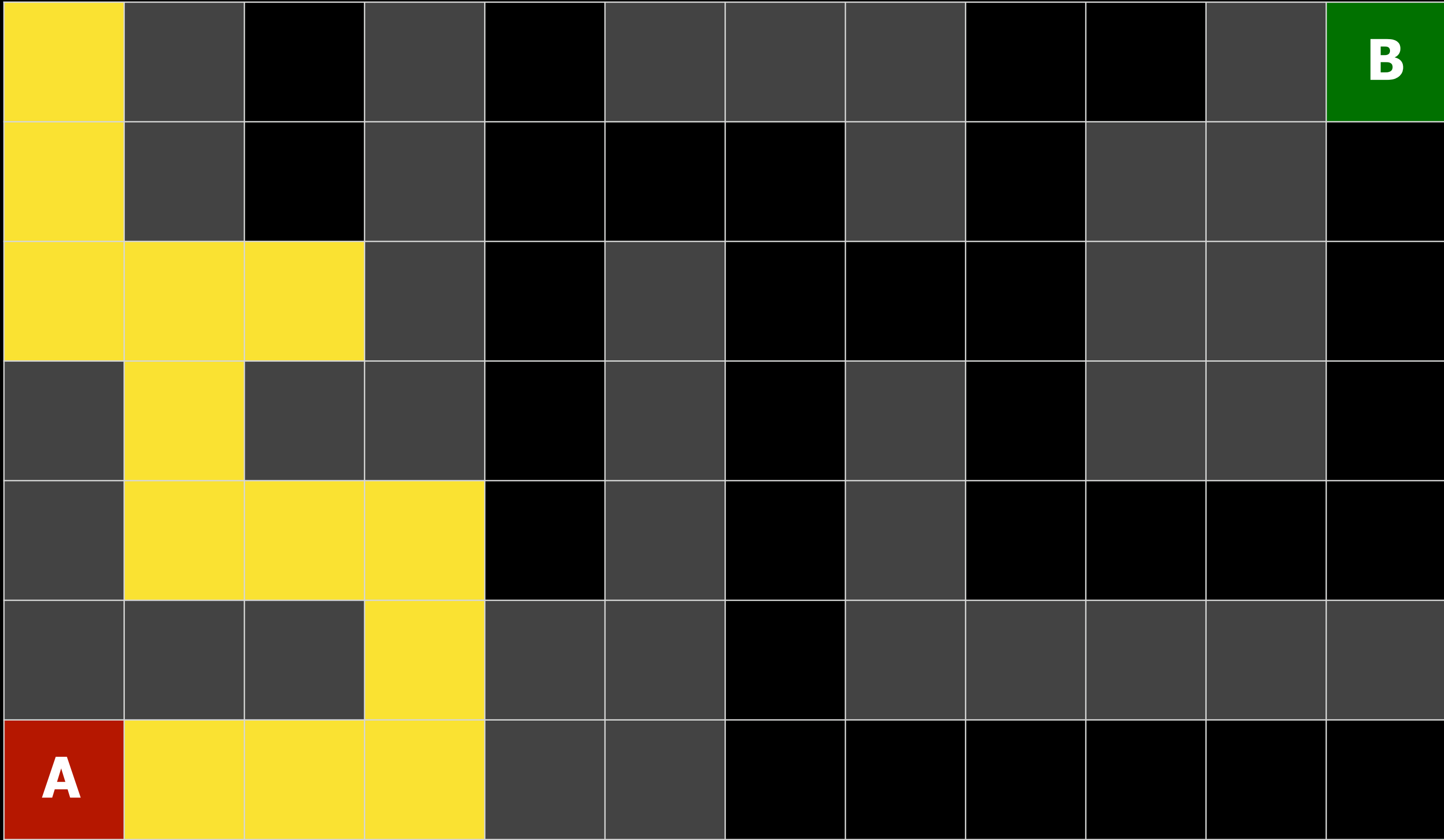
Depth-First Search



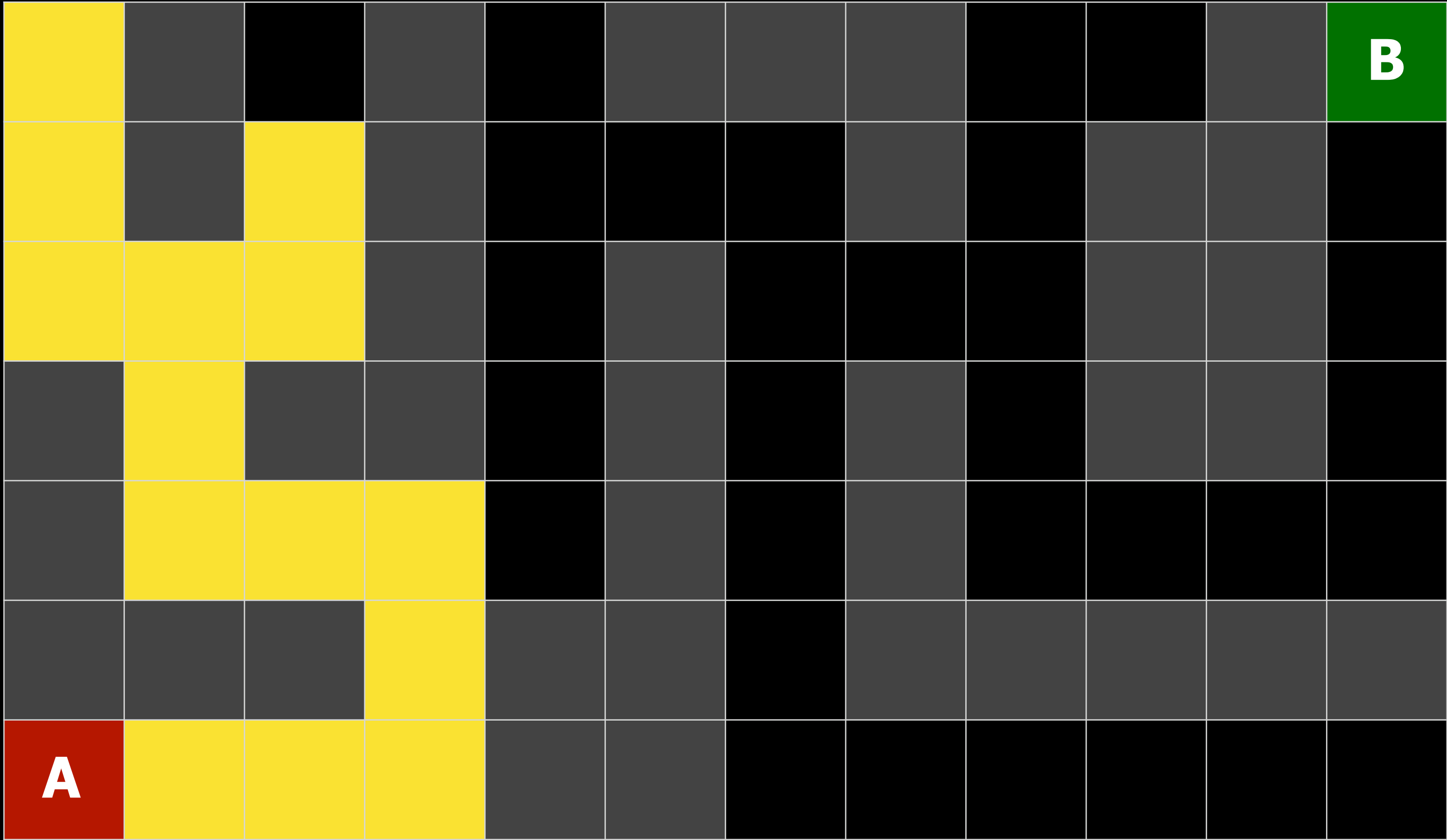
Depth-First Search



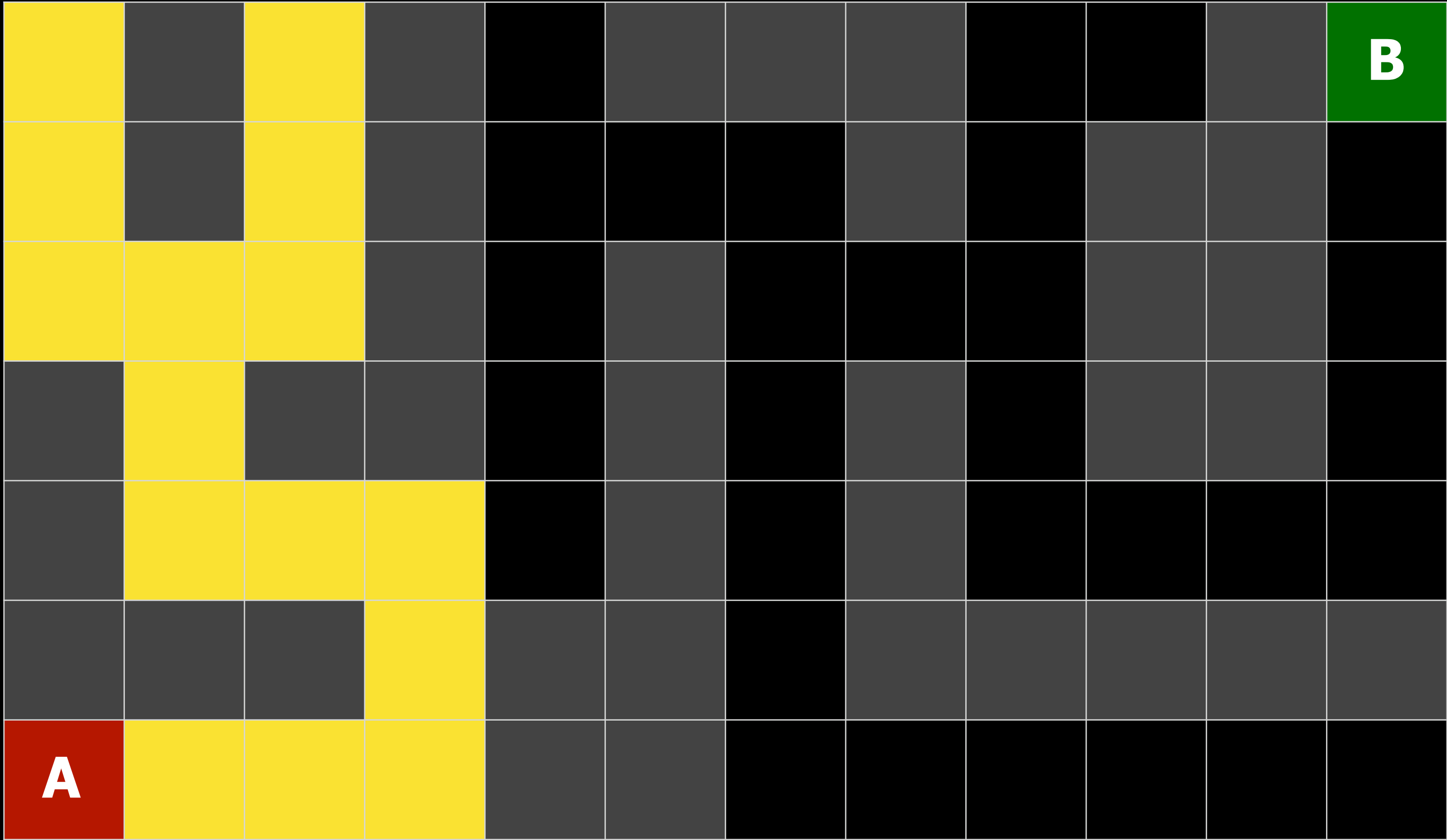
Depth-First Search



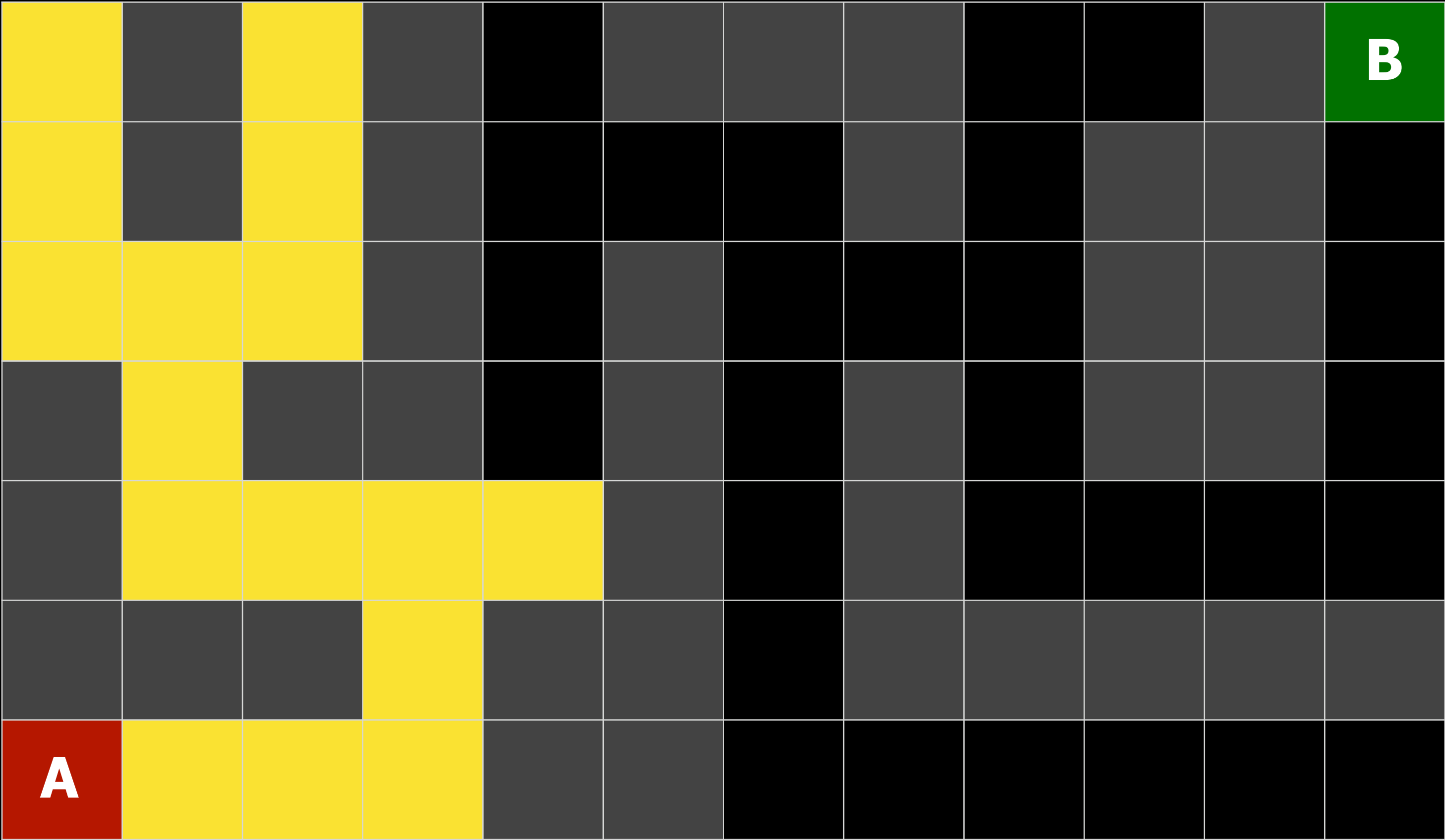
Depth-First Search



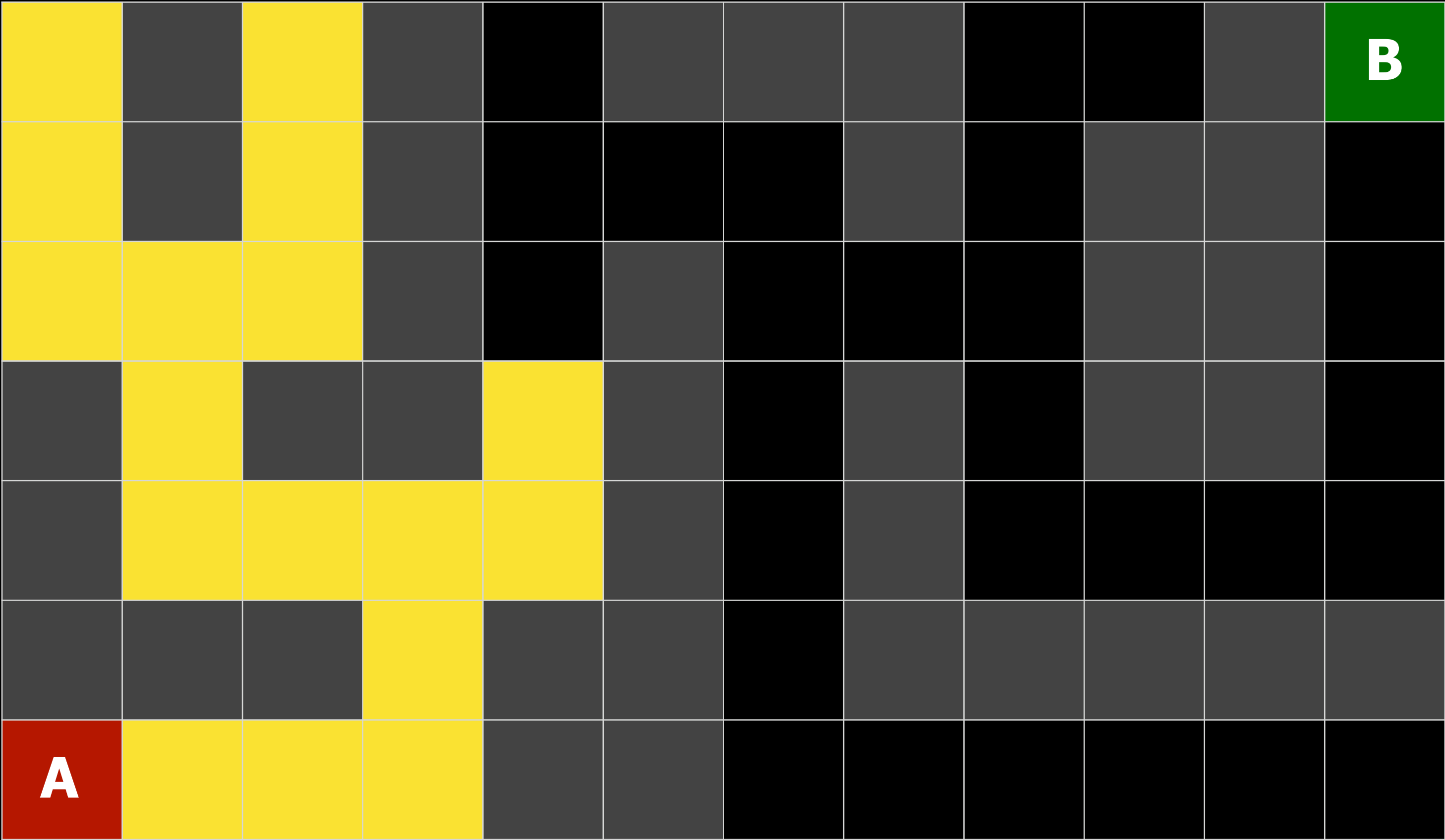
Depth-First Search



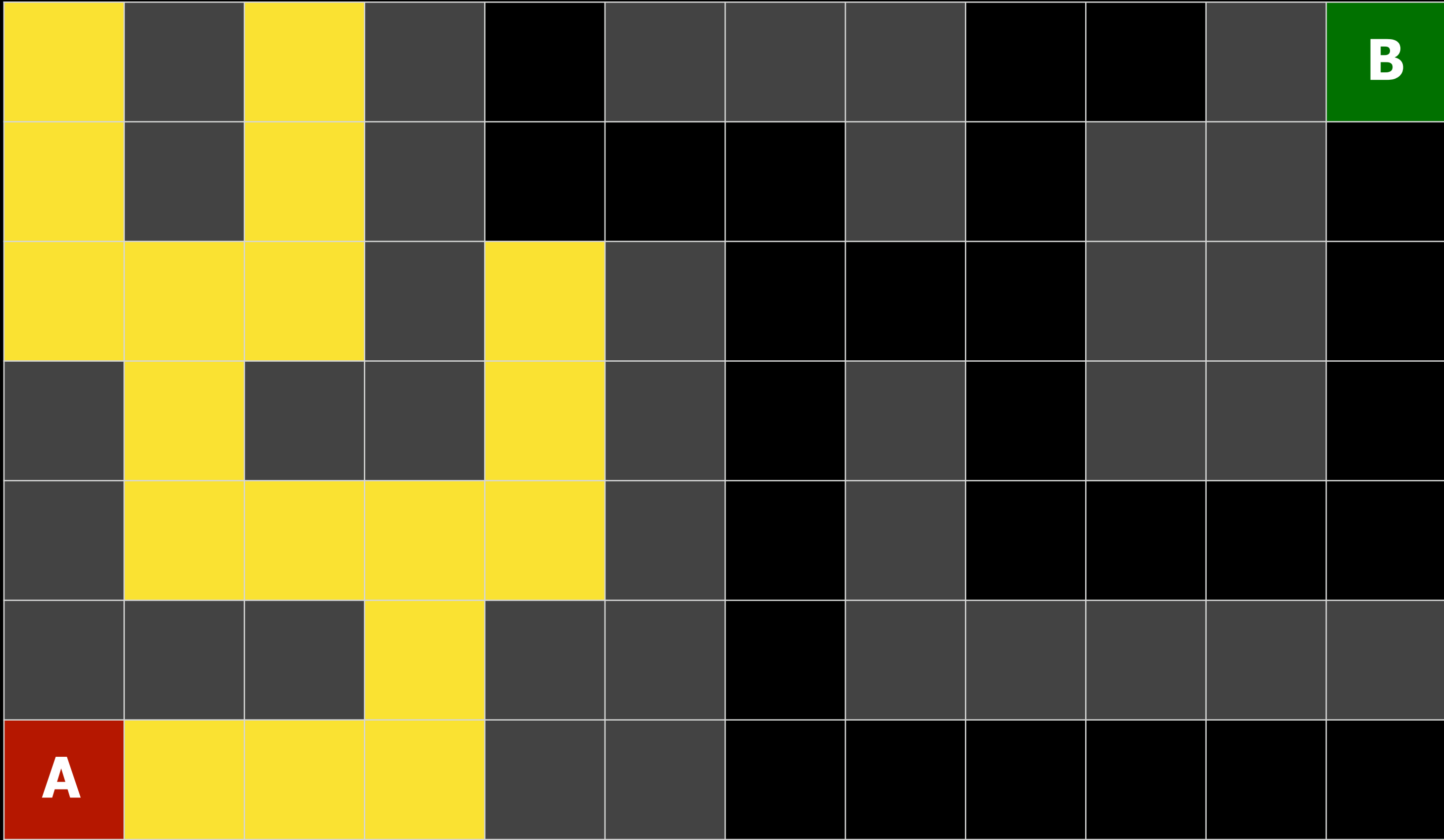
Depth-First Search



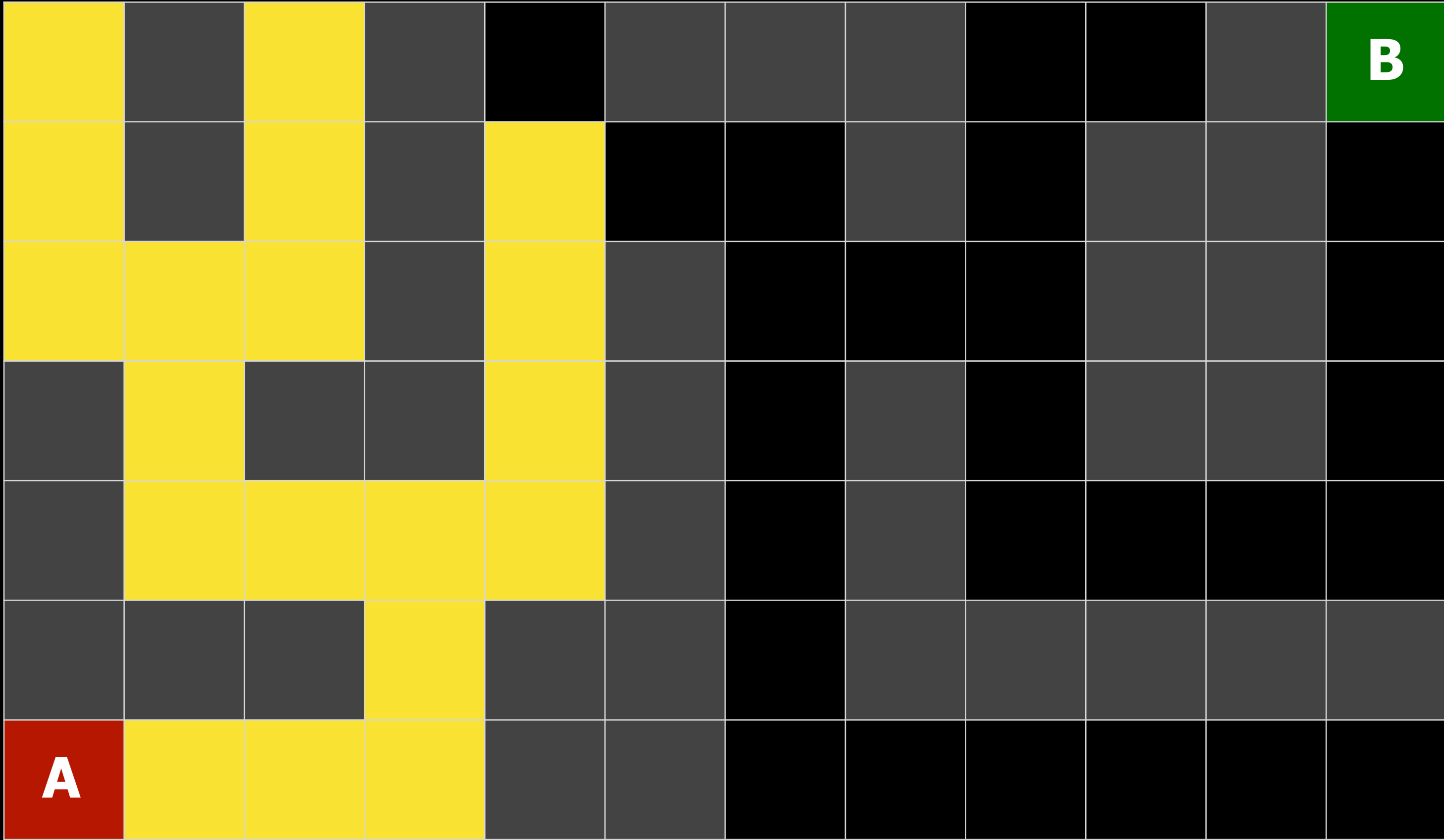
Depth-First Search



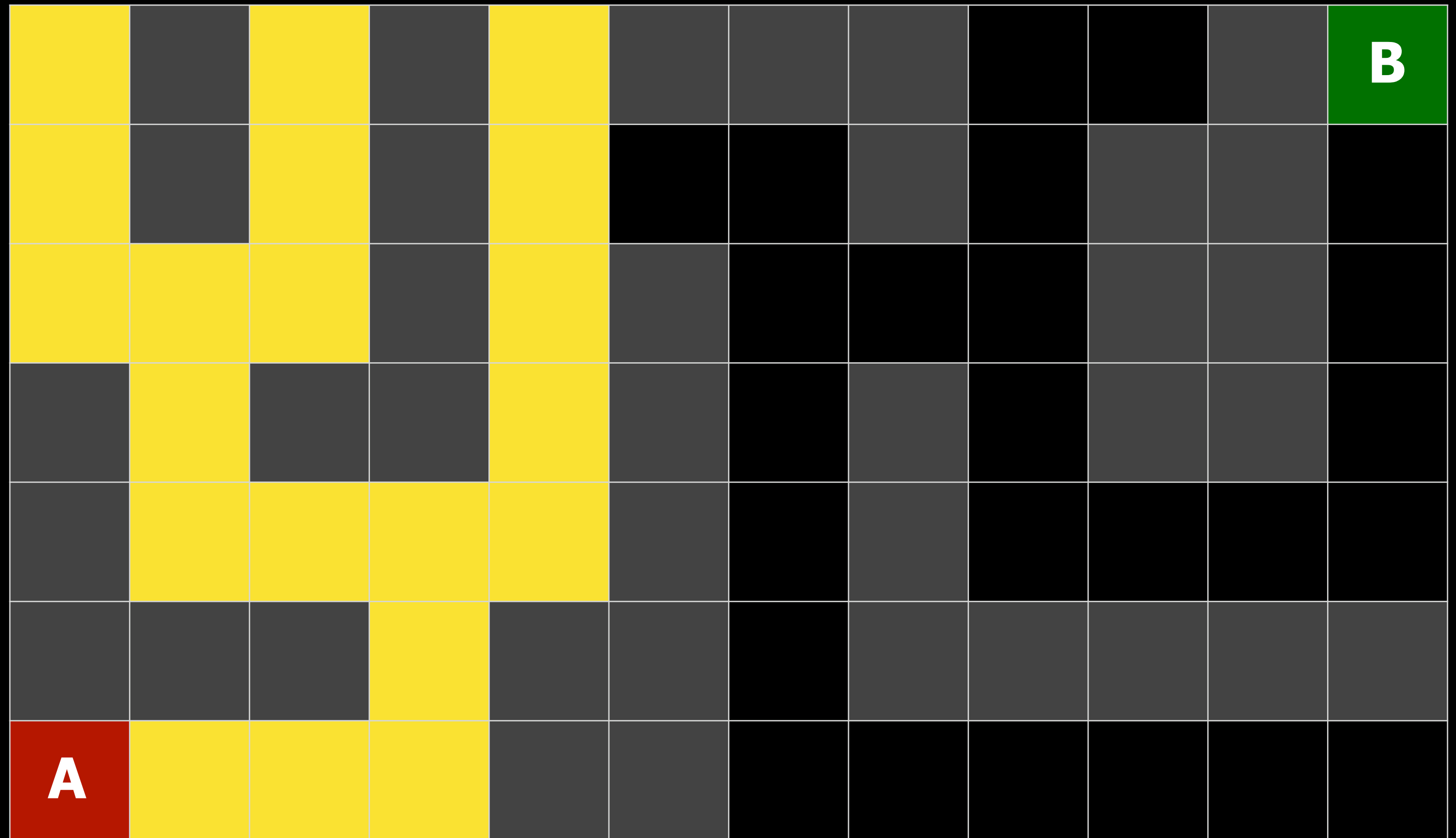
Depth-First Search



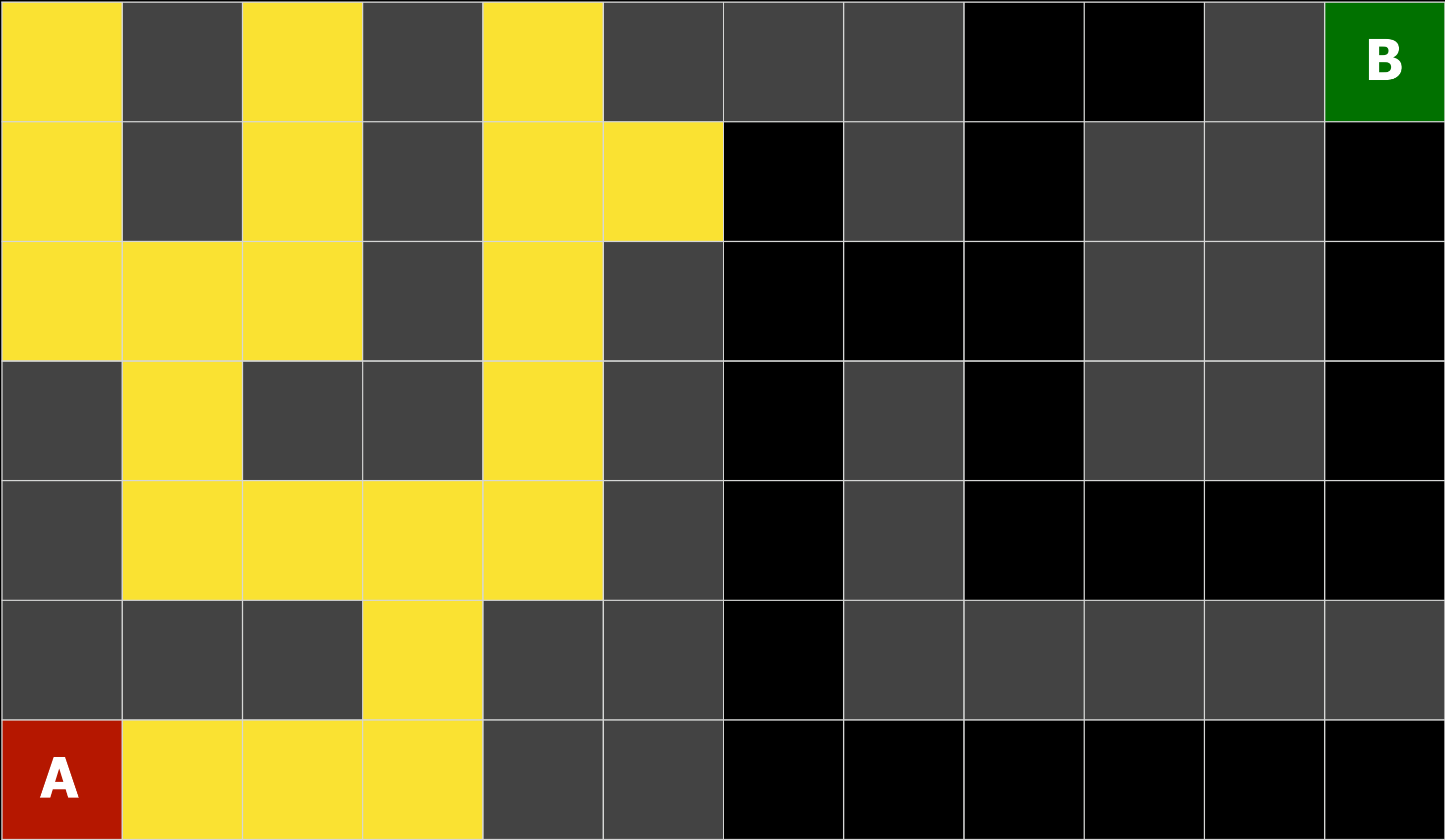
Depth-First Search



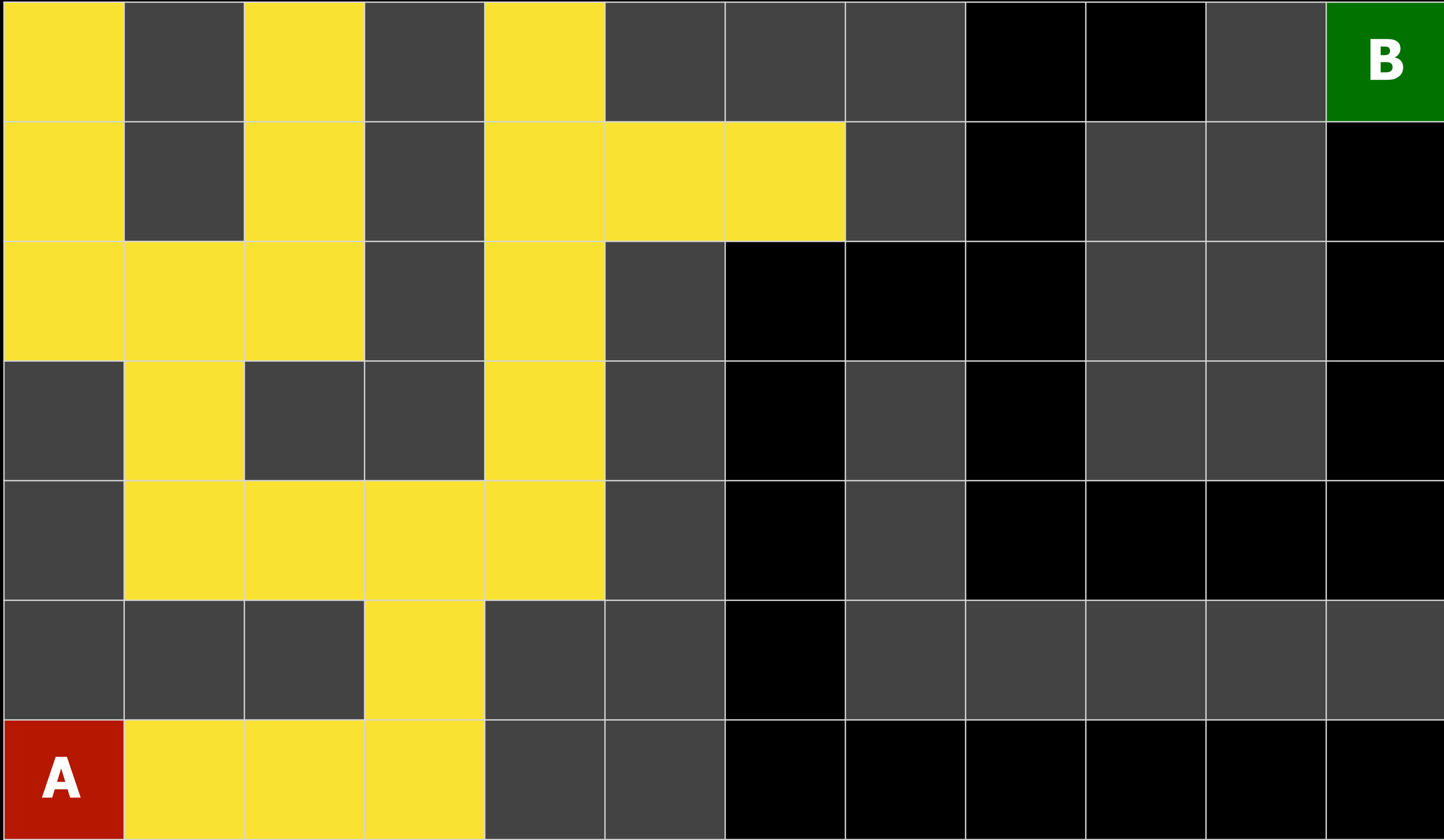
Depth-First Search



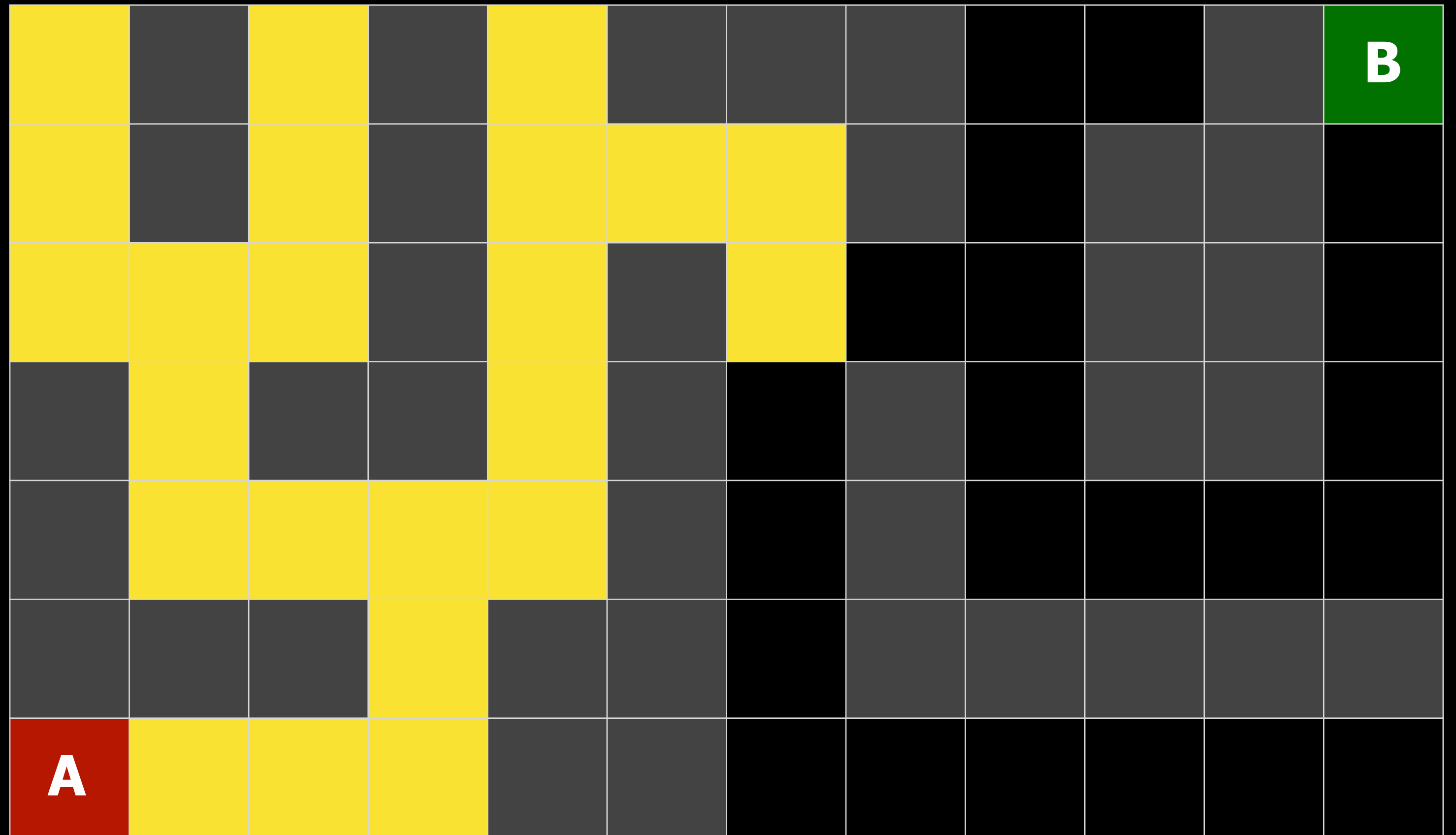
Depth-First Search



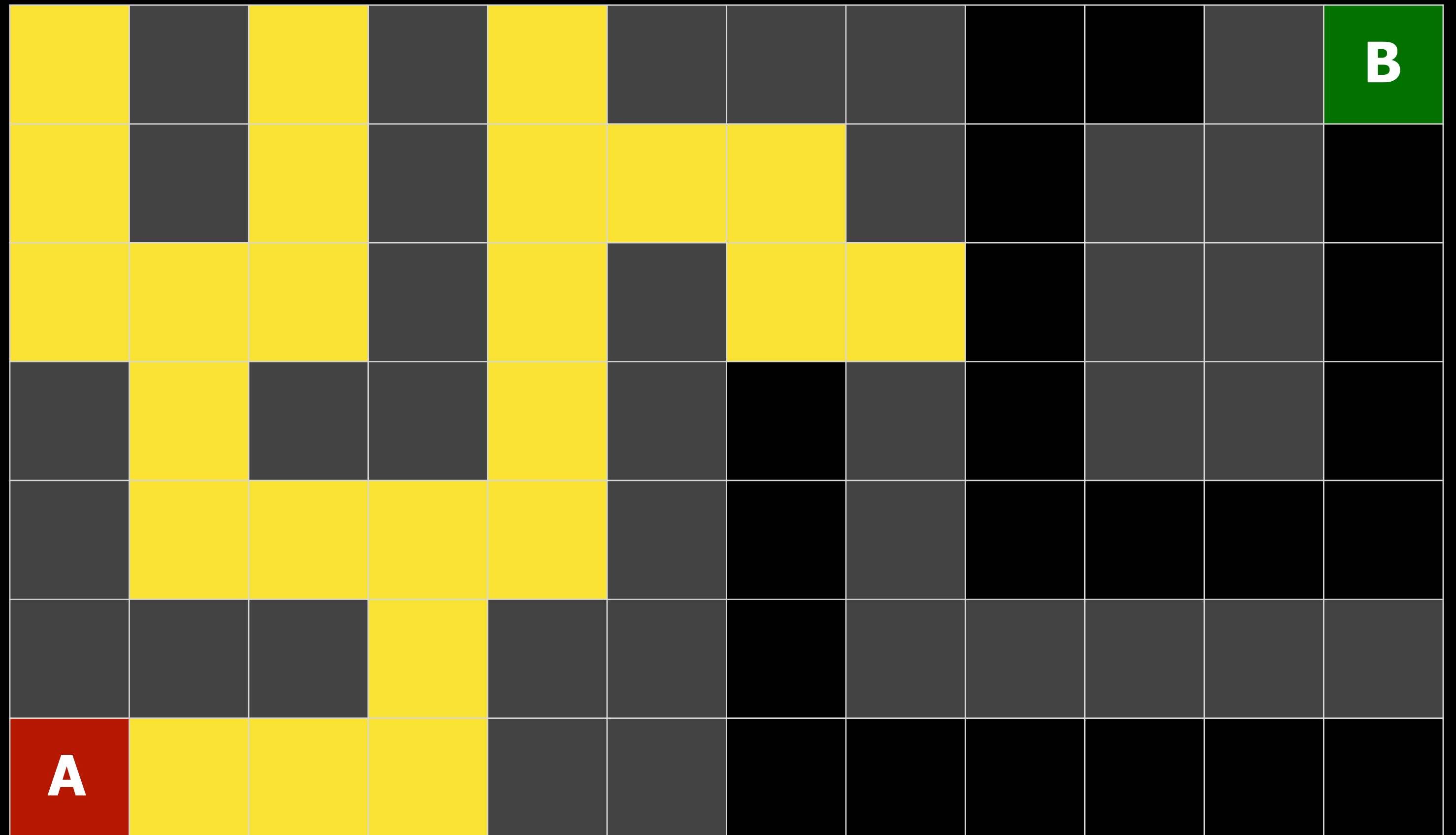
Depth-First Search



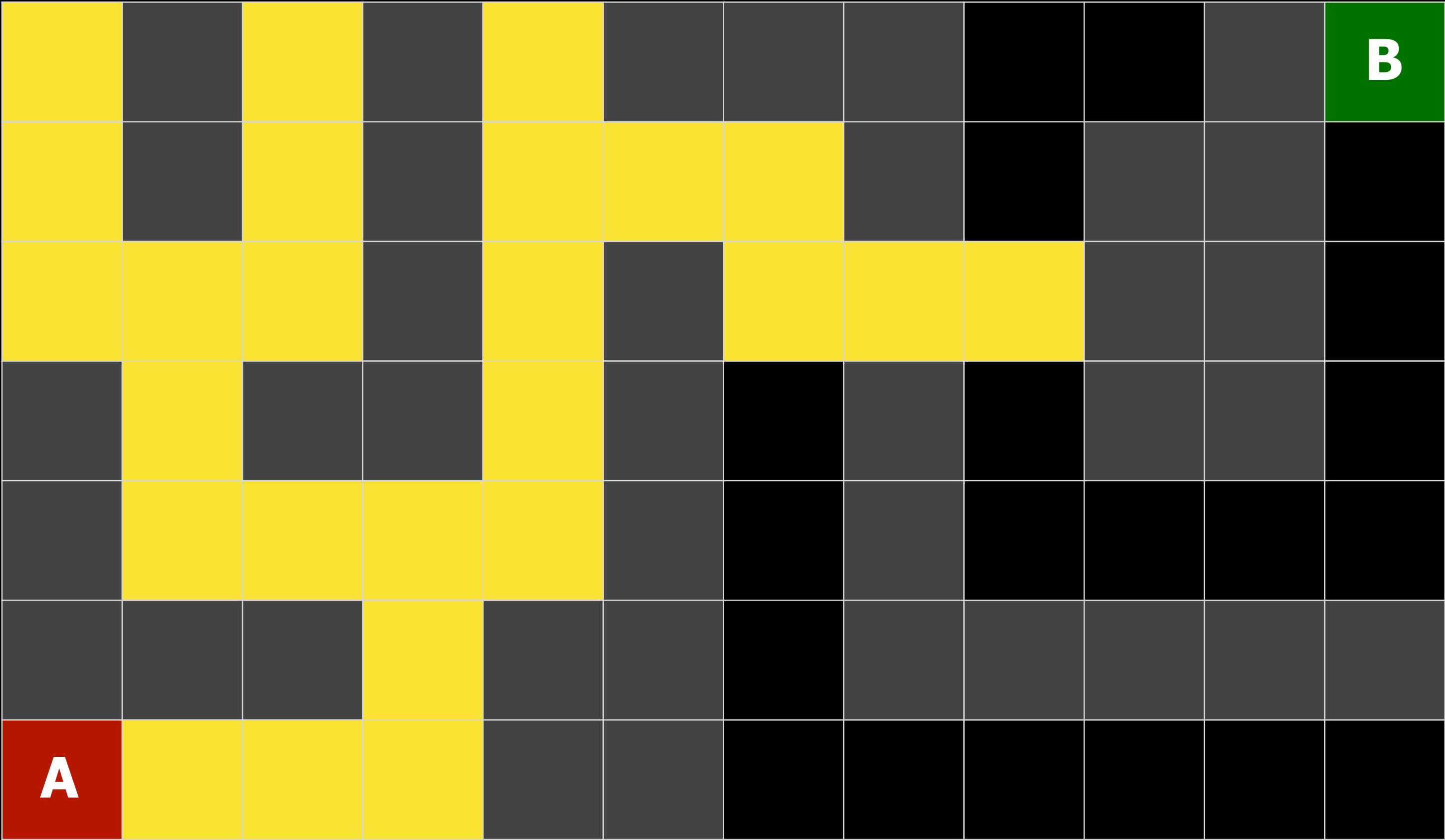
Depth-First Search



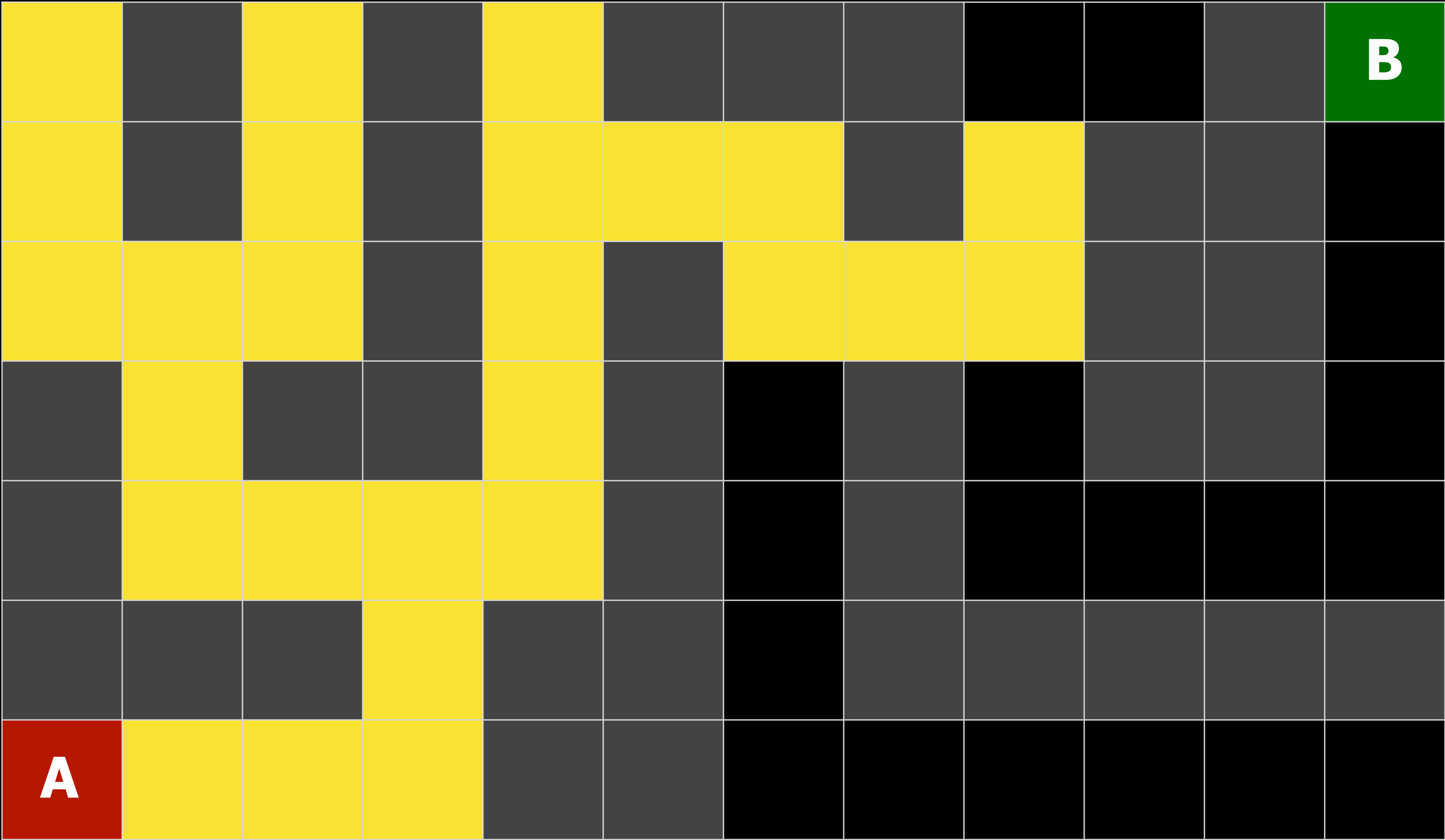
Depth-First Search



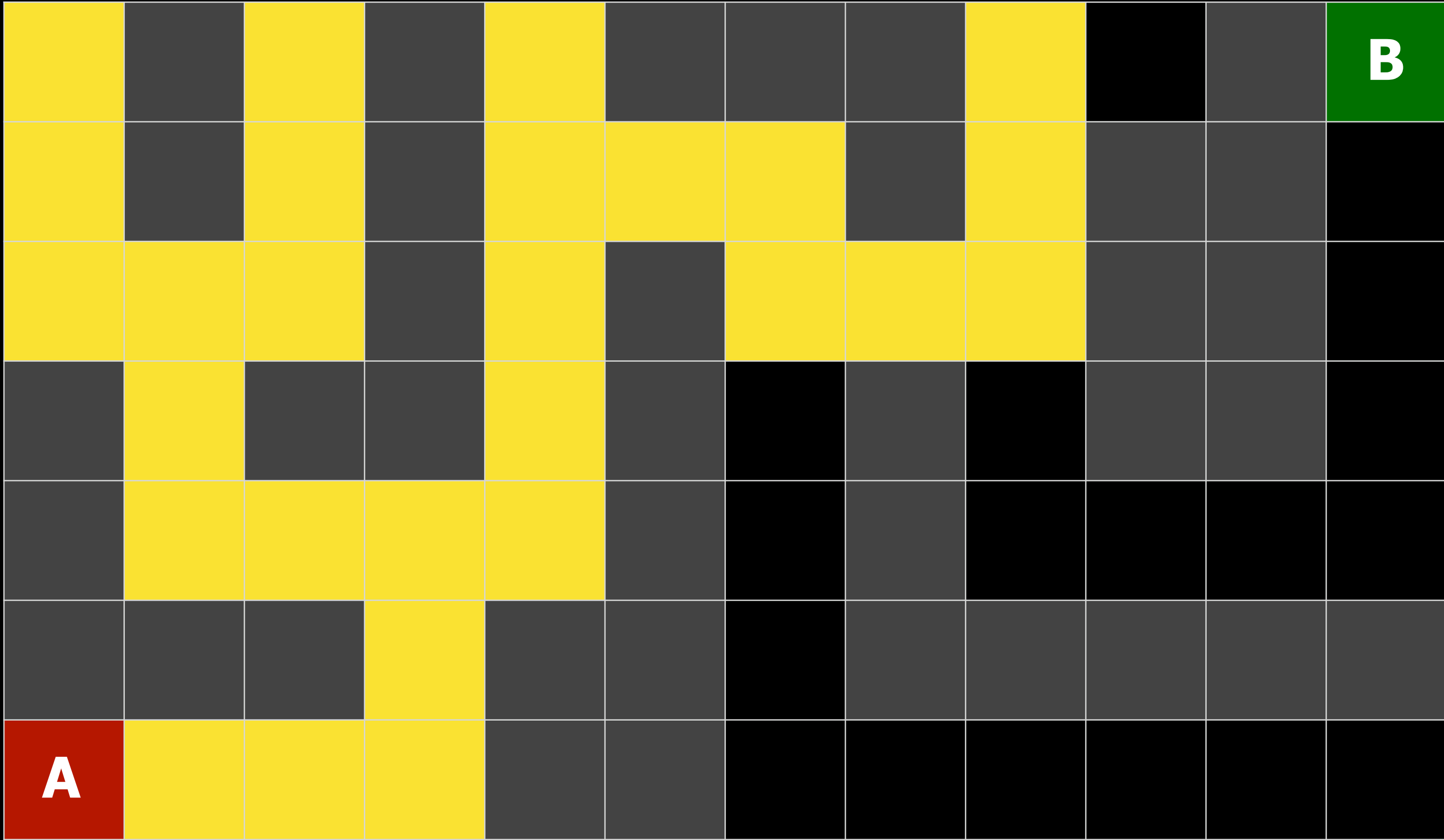
Depth-First Search



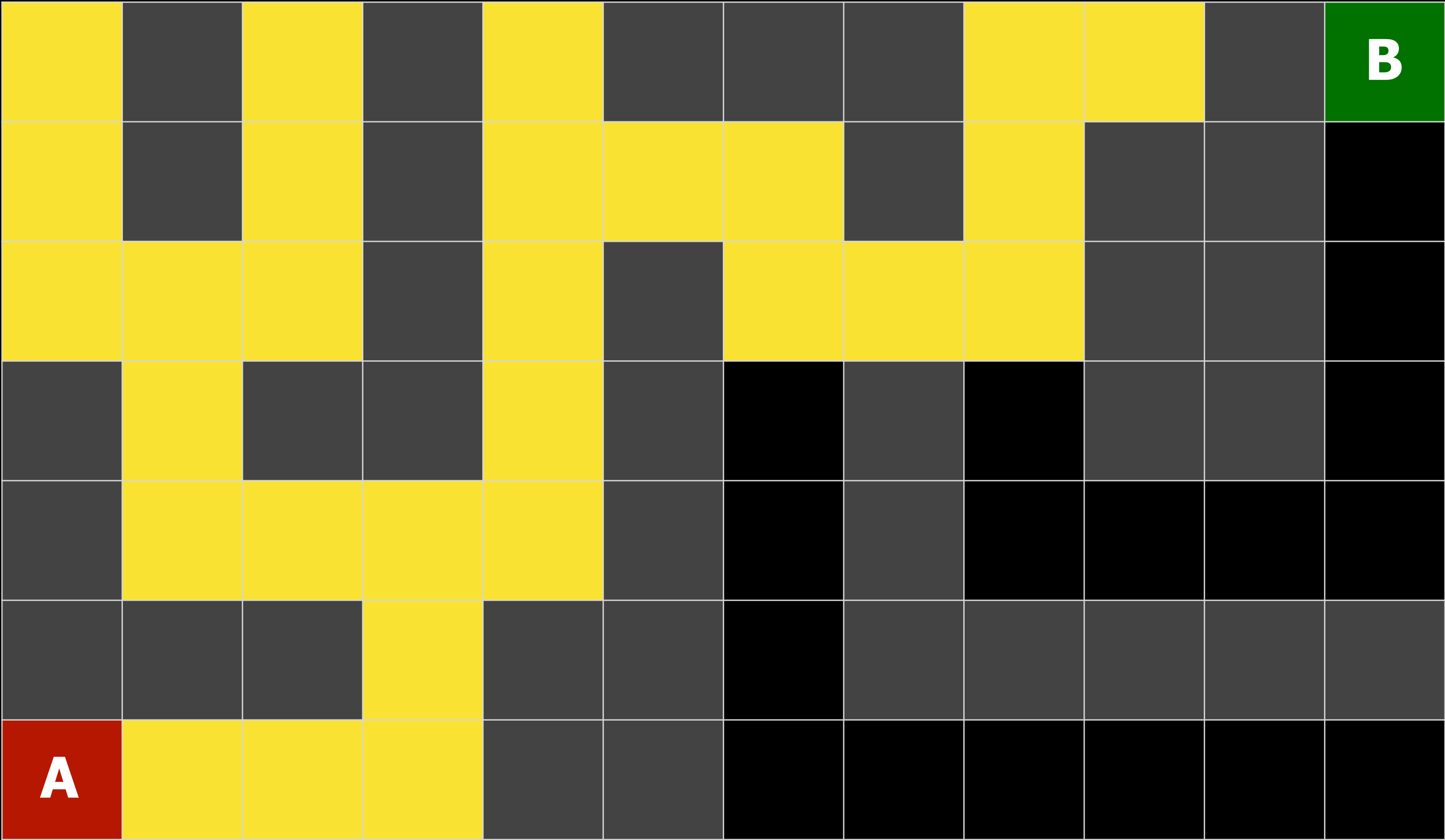
Depth-First Search



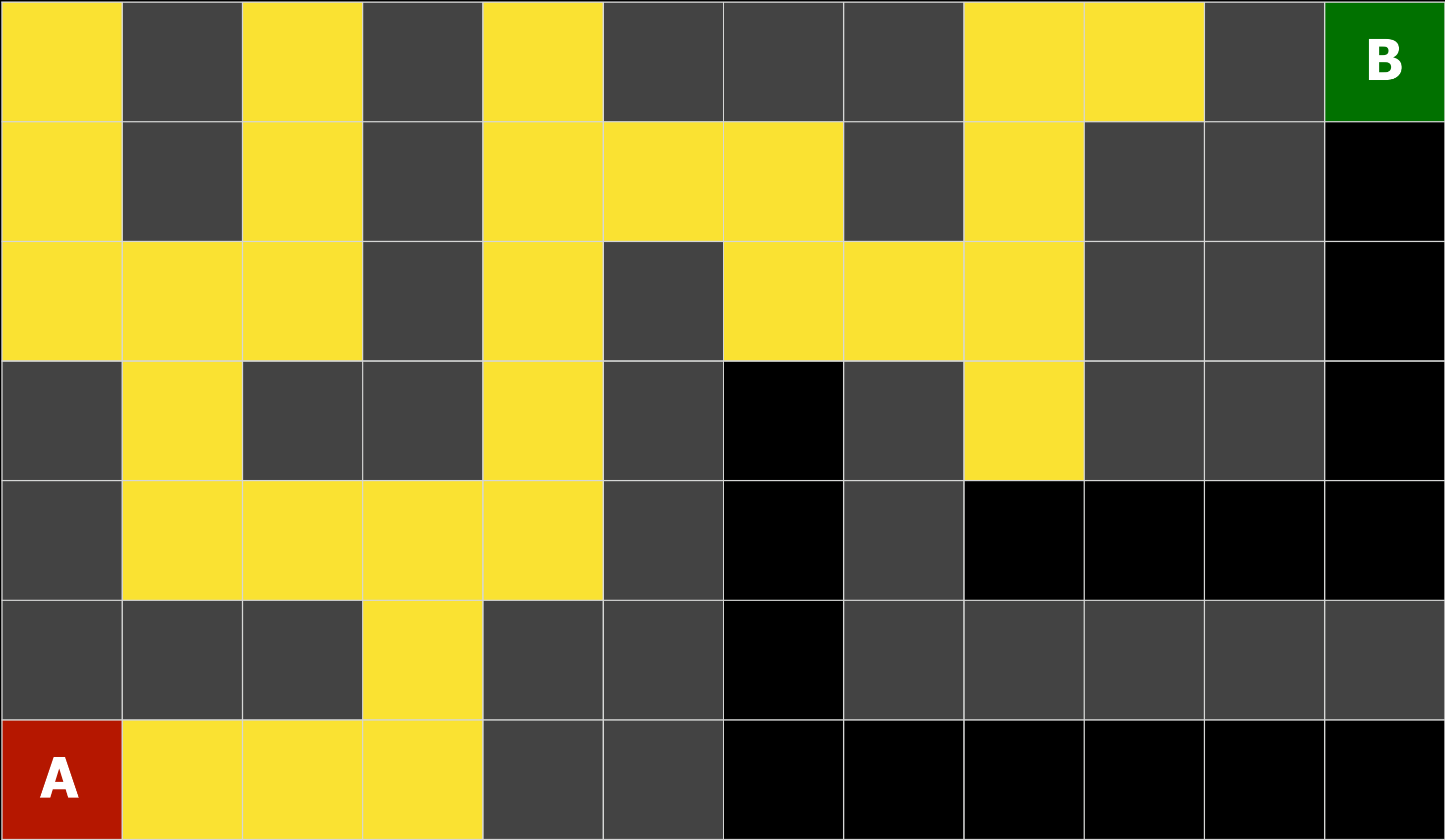
Depth-First Search



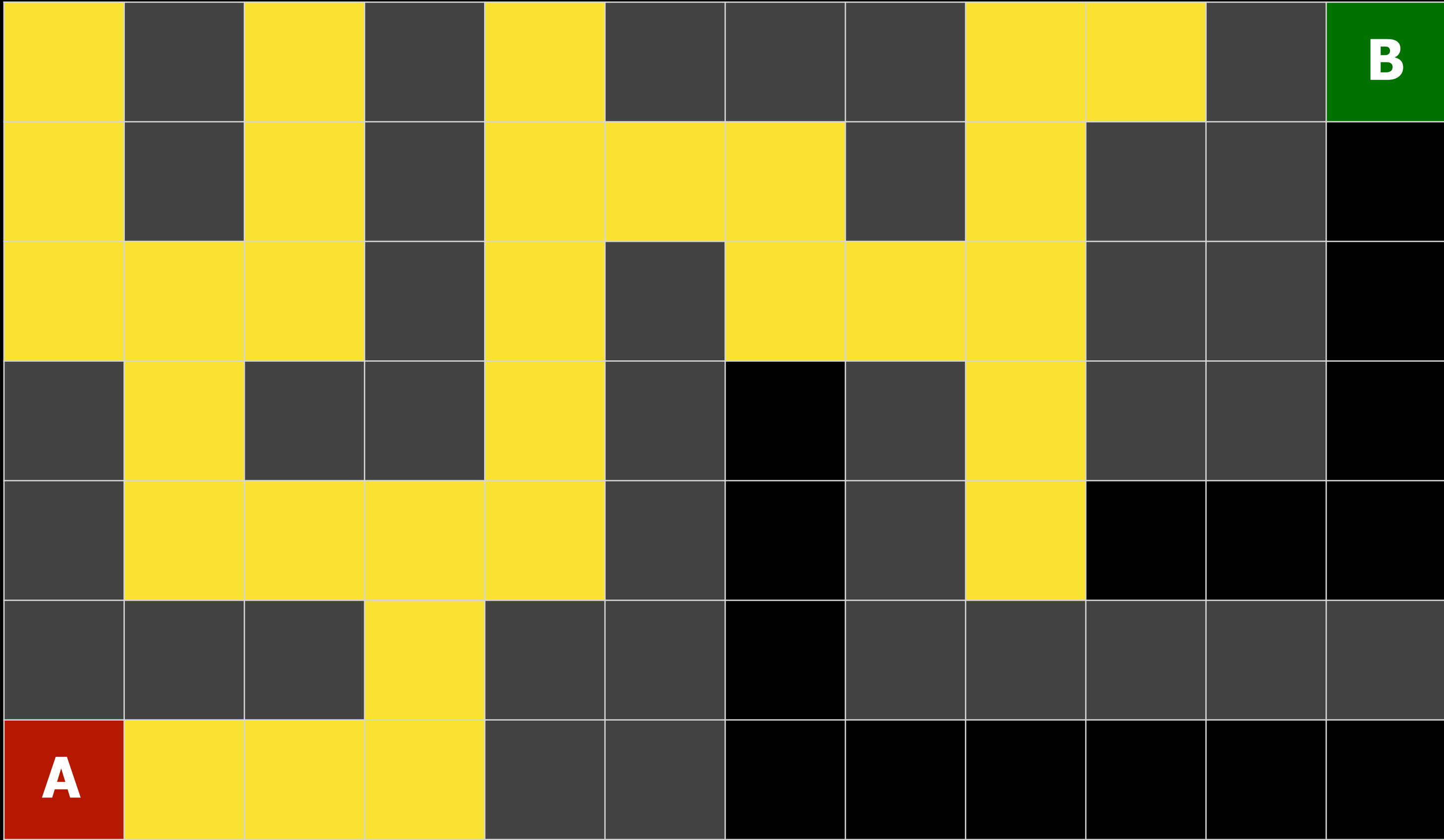
Depth-First Search



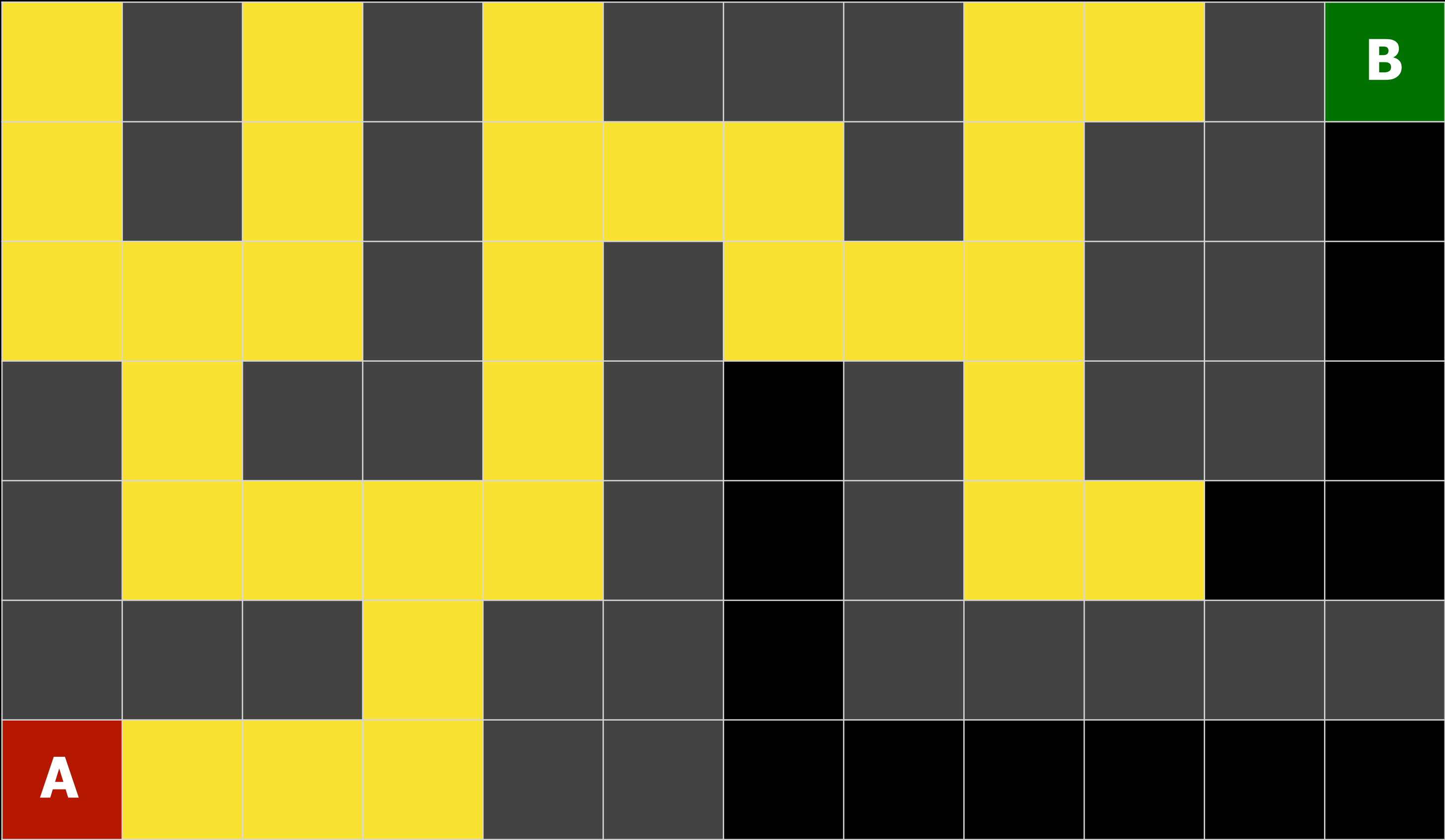
Depth-First Search



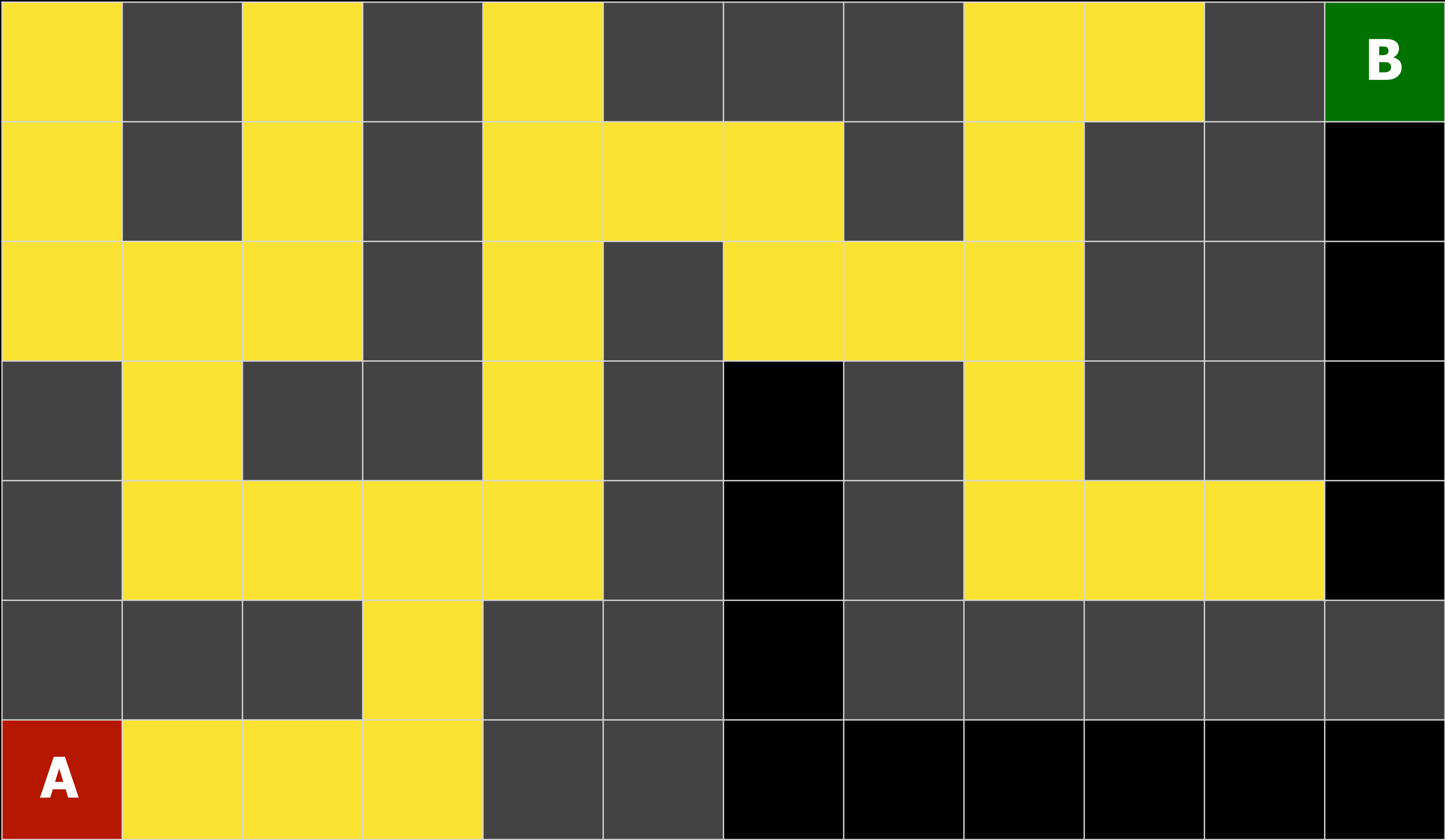
Depth-First Search



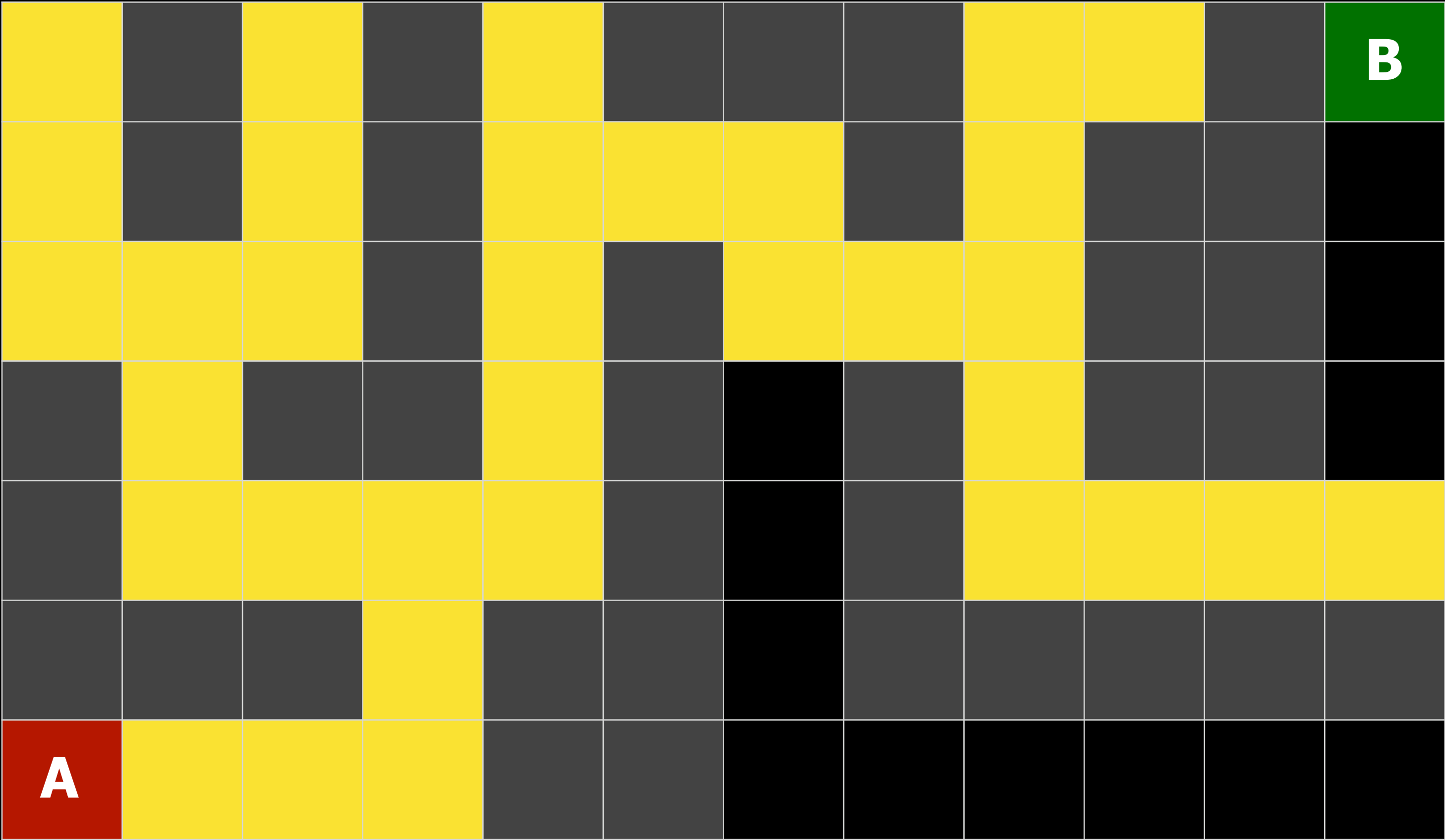
Depth-First Search



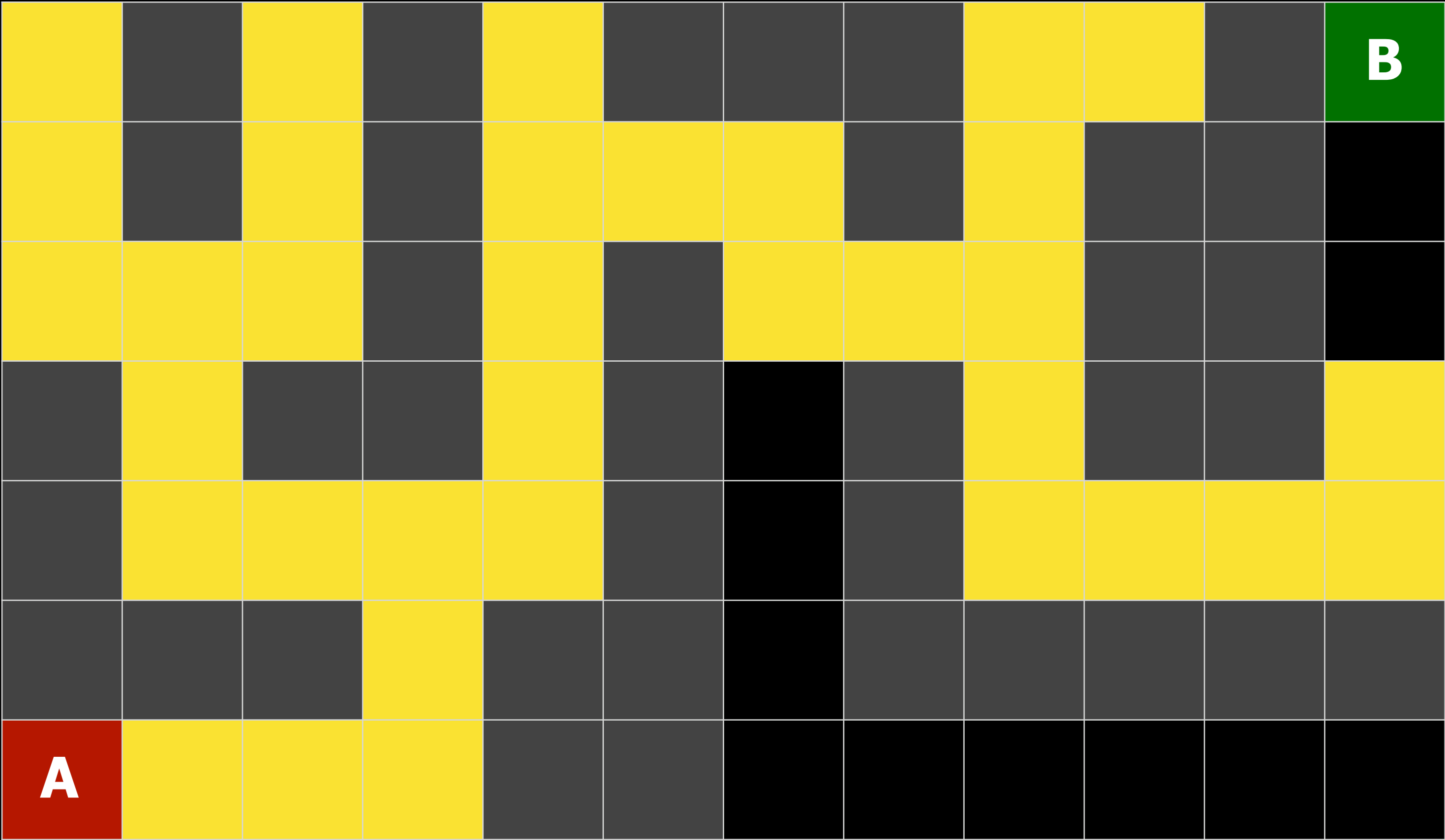
Depth-First Search



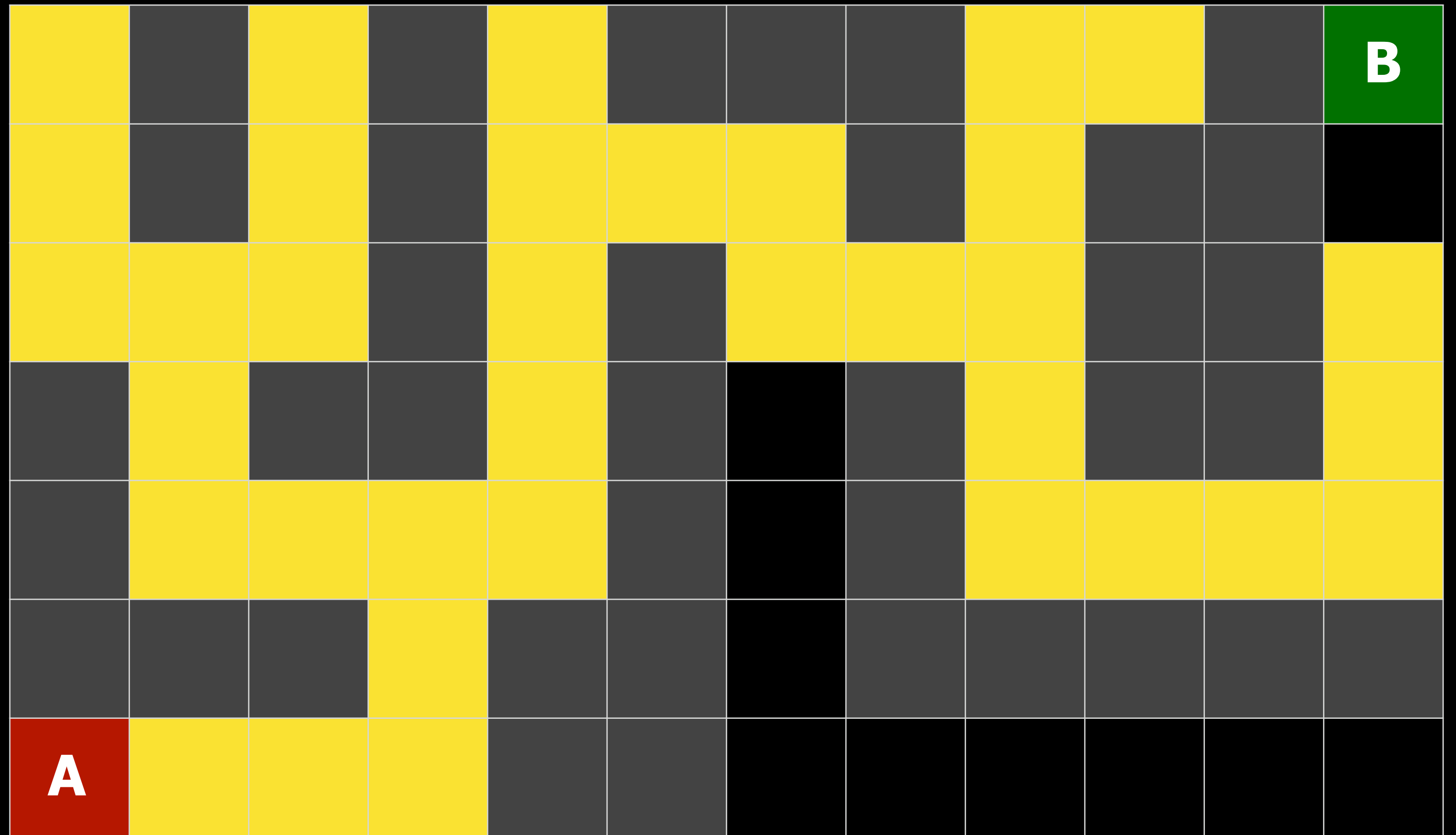
Depth-First Search



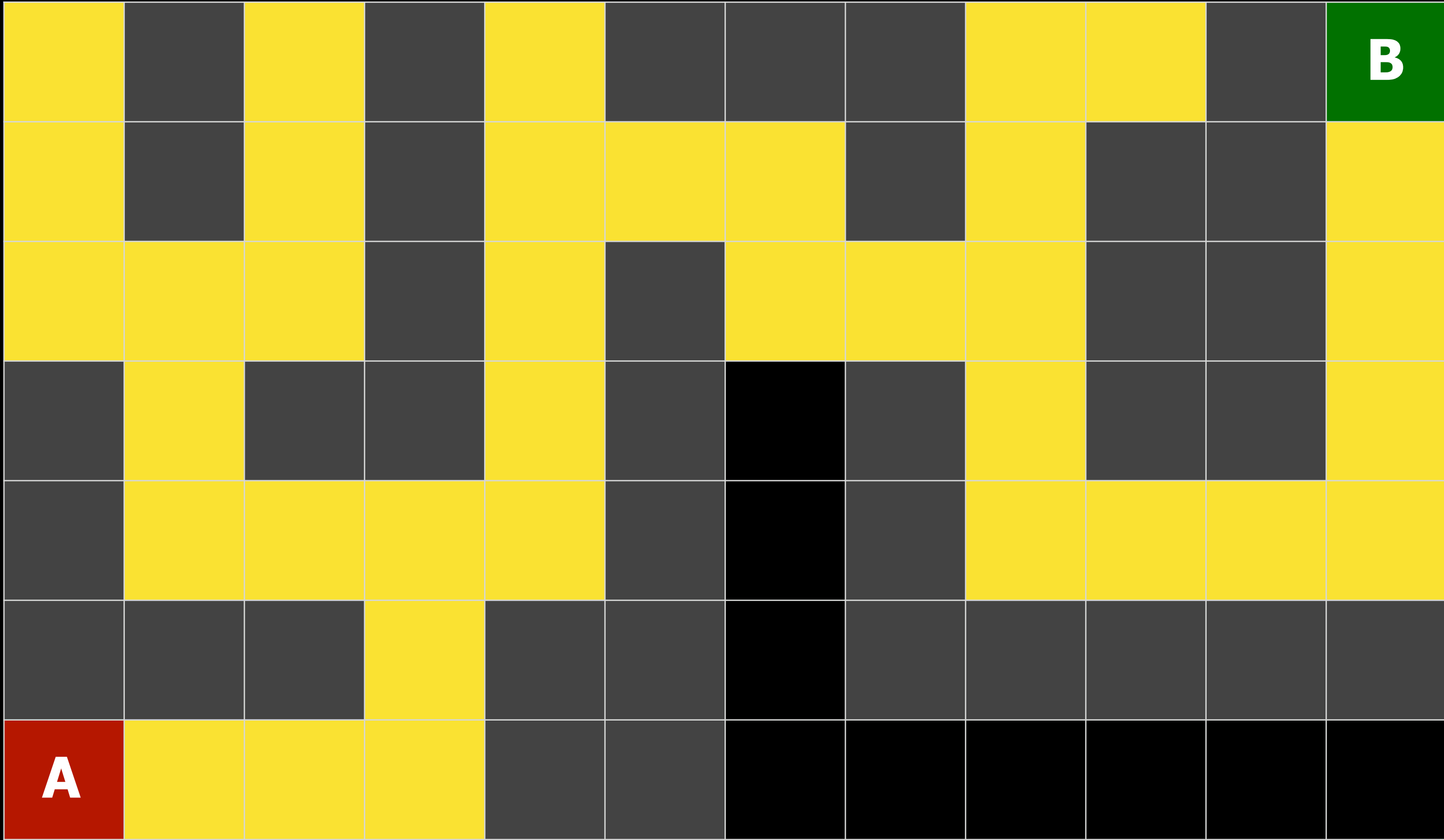
Depth-First Search



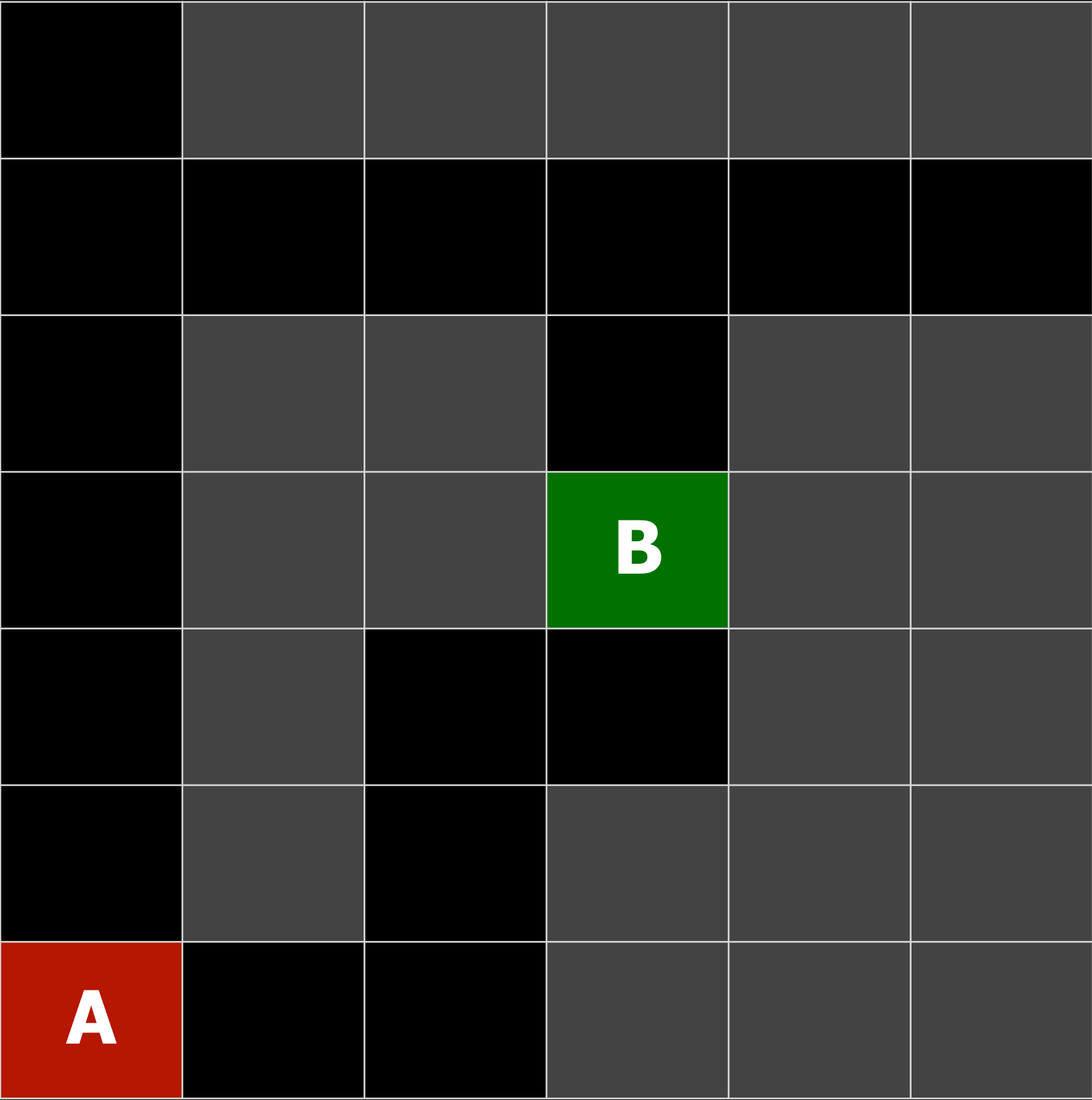
Depth-First Search



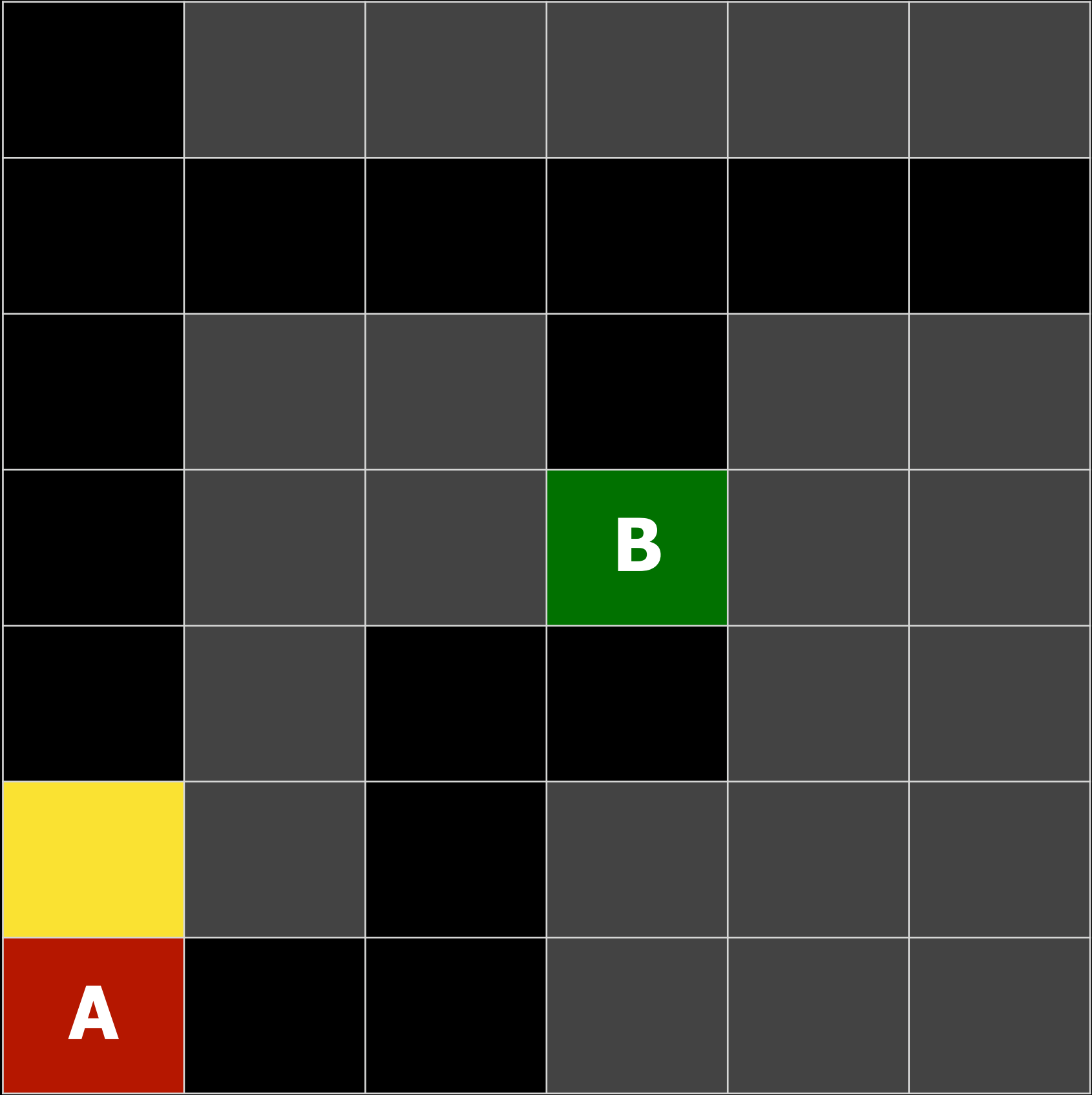
Depth-First Search



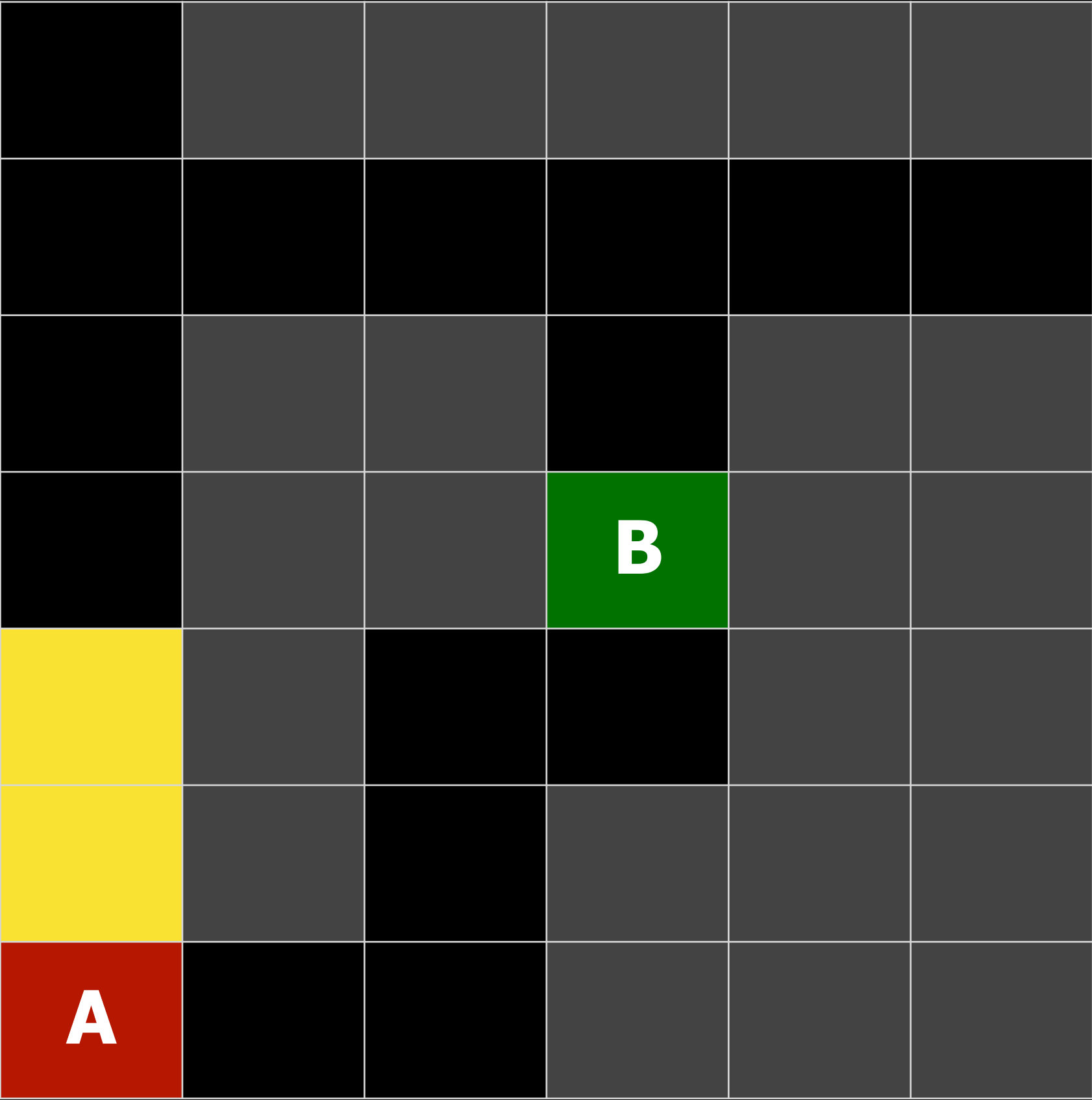
Depth-First Search



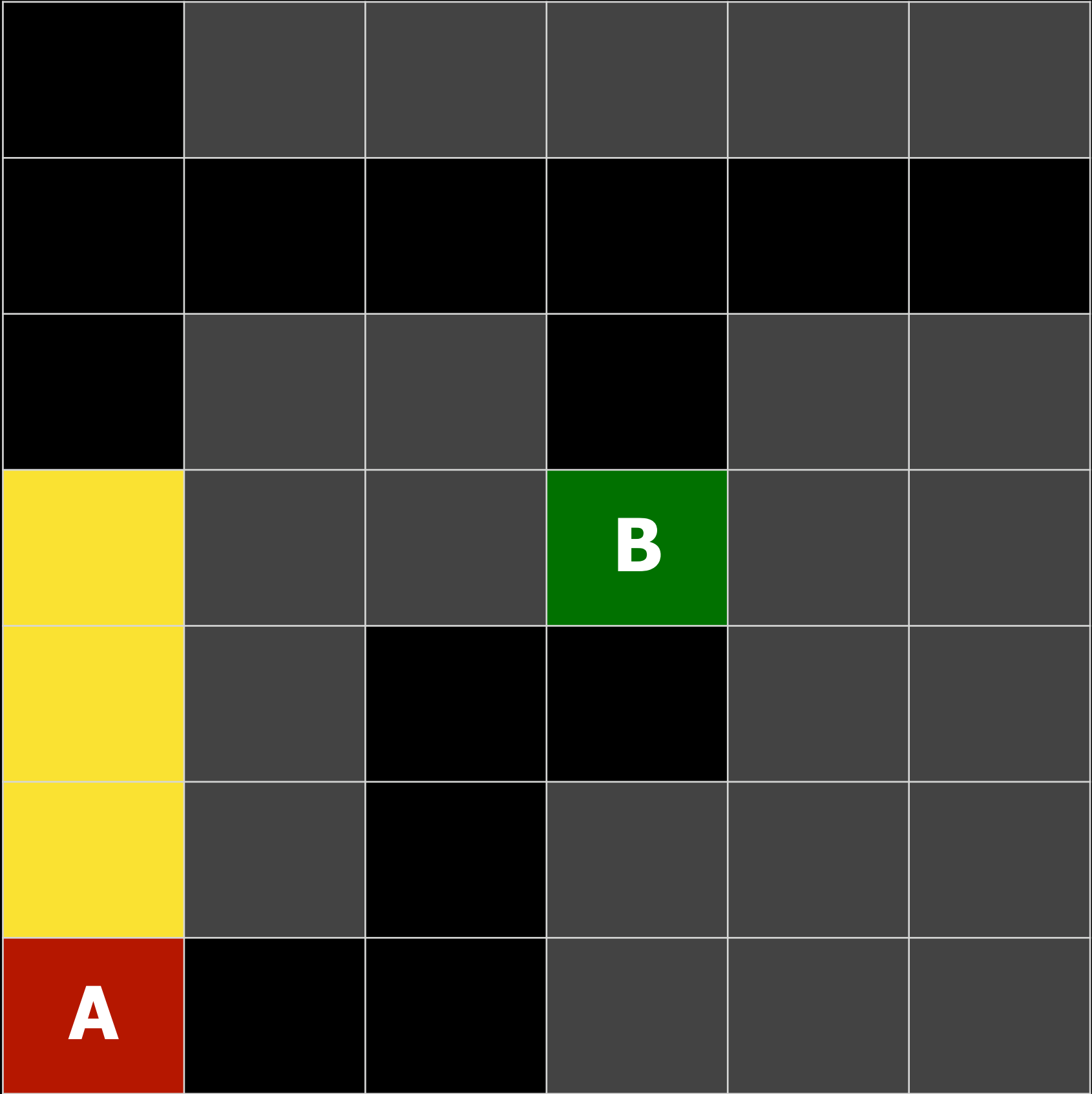
Depth-First Search



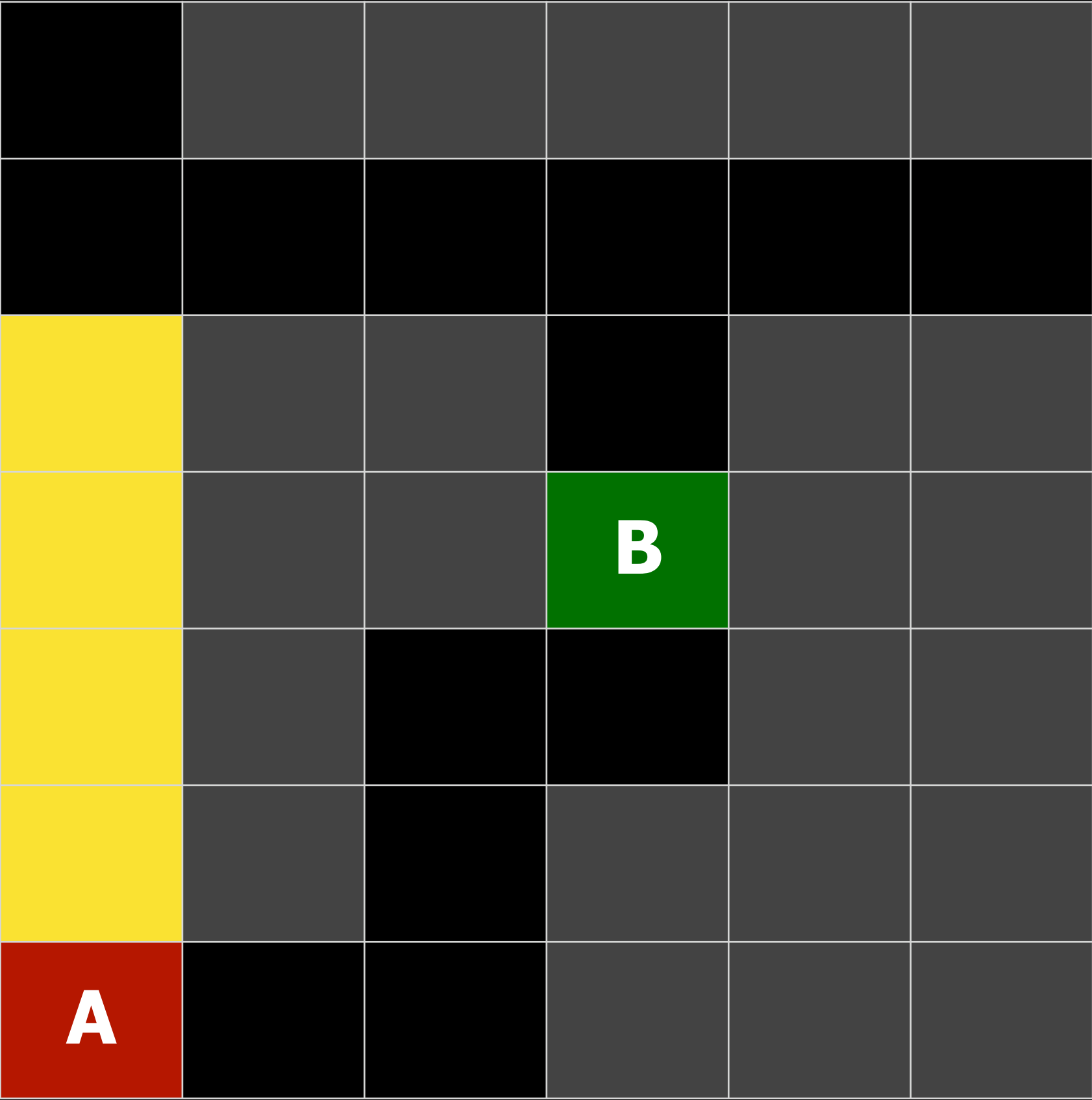
Depth-First Search



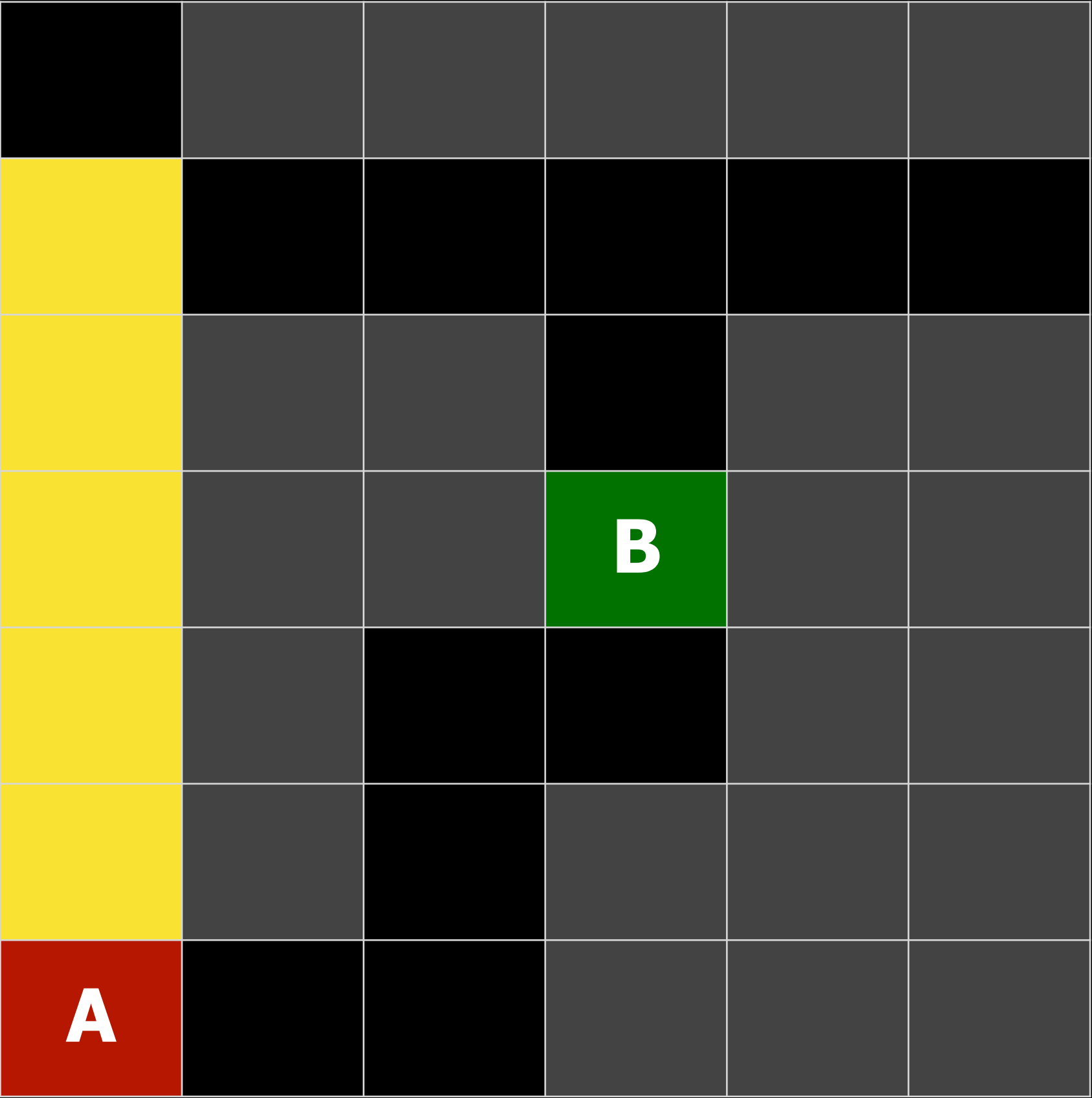
Depth-First Search



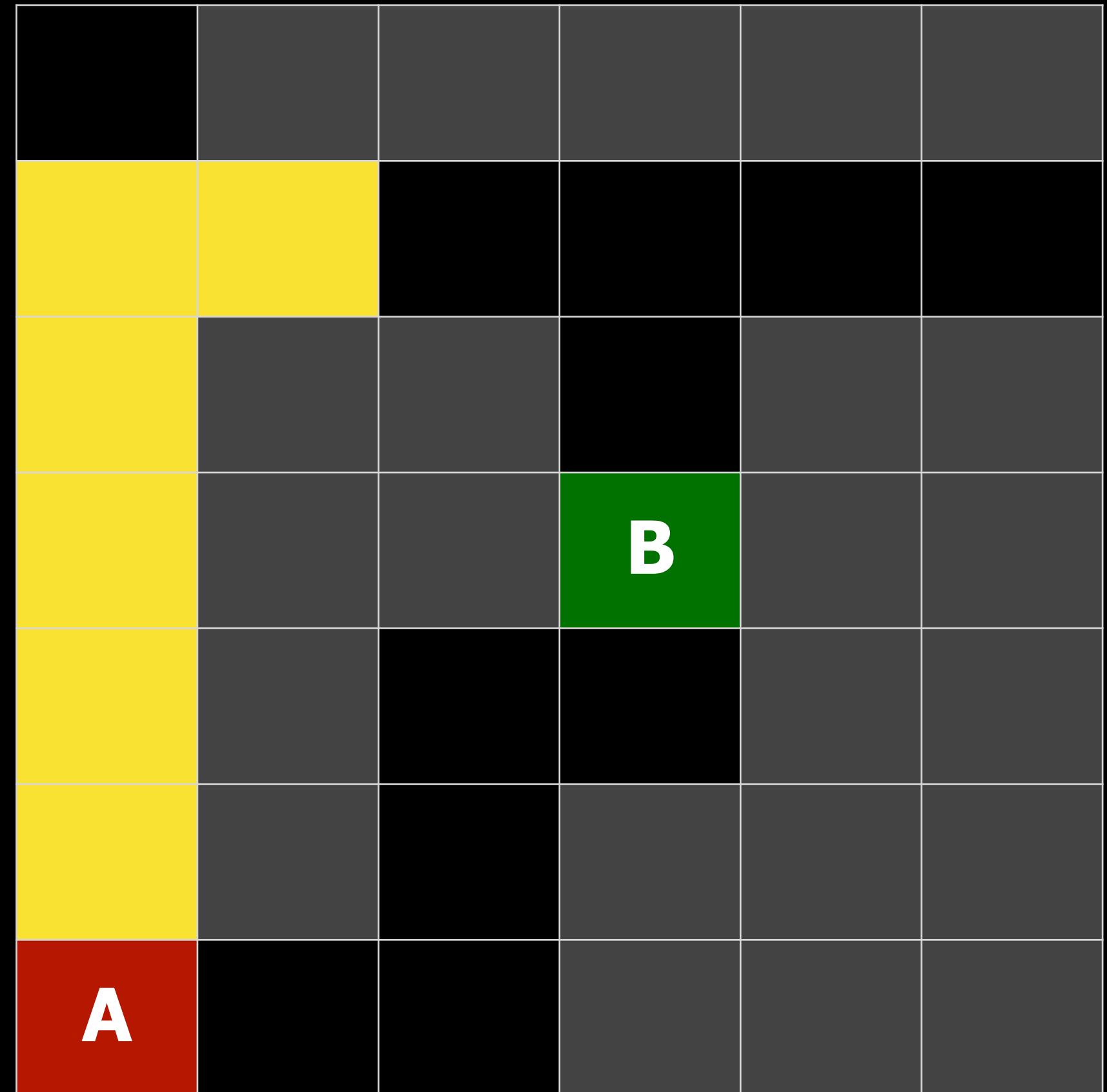
Depth-First Search



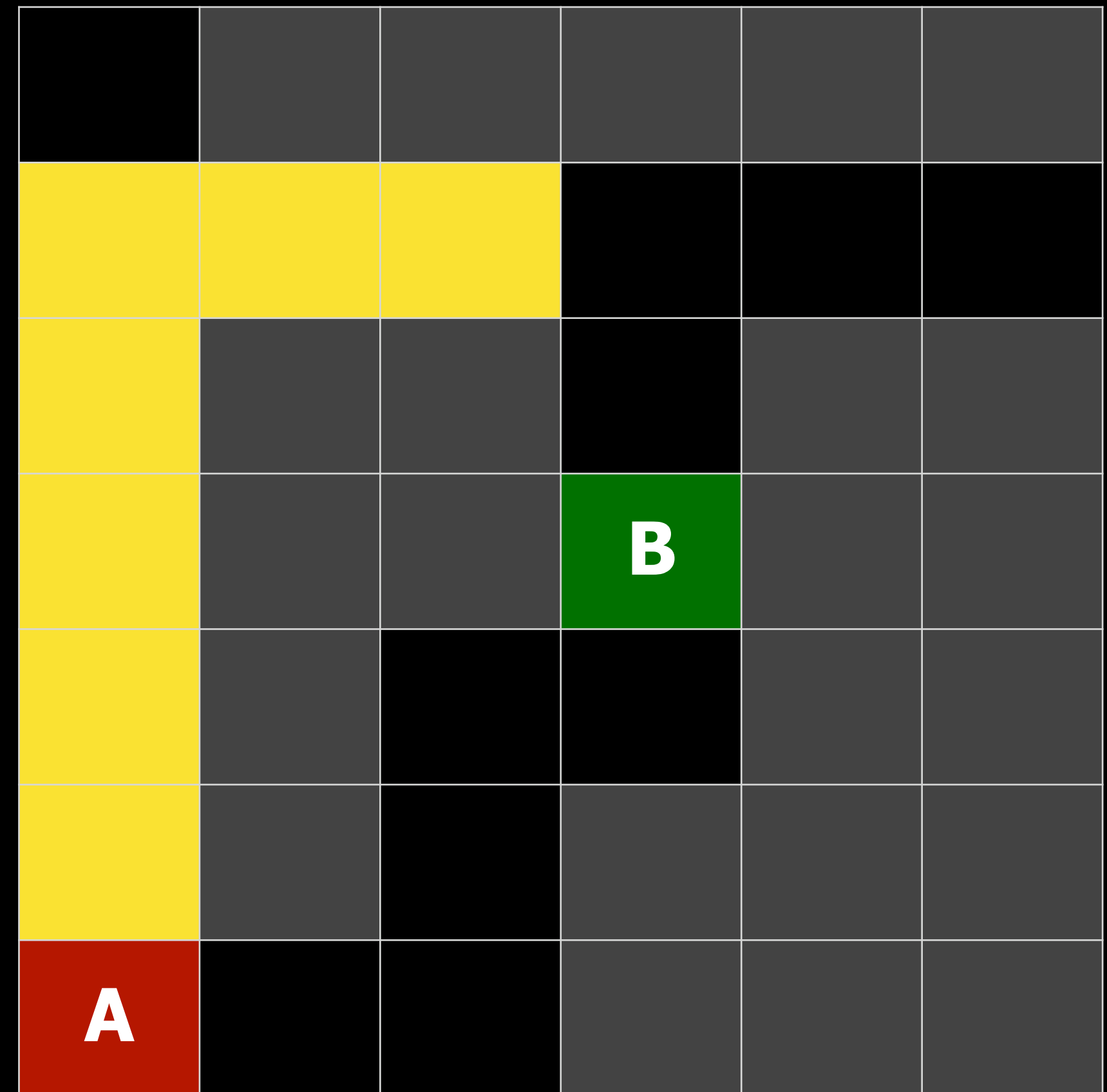
Depth-First Search



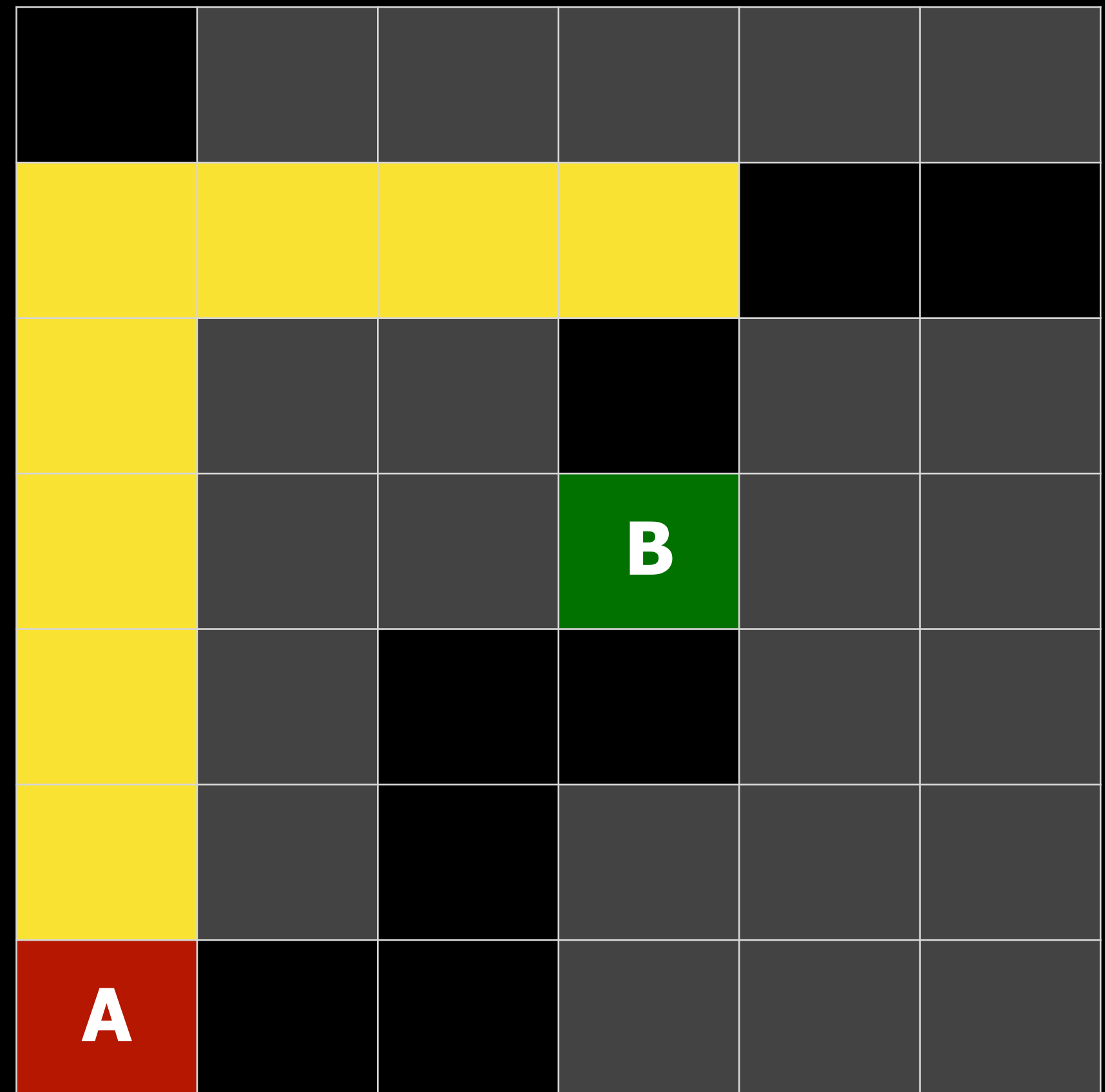
Depth-First Search



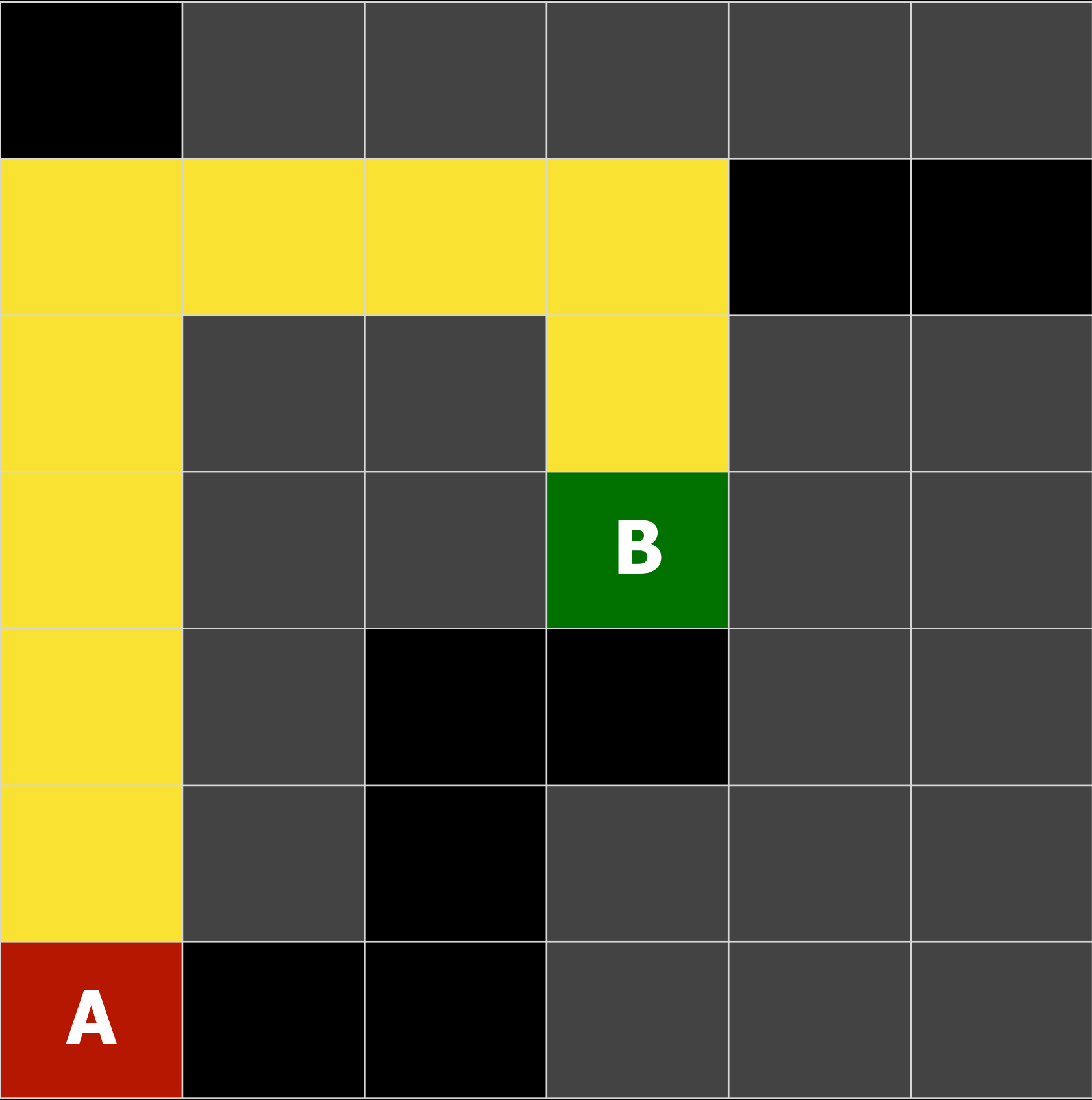
Depth-First Search



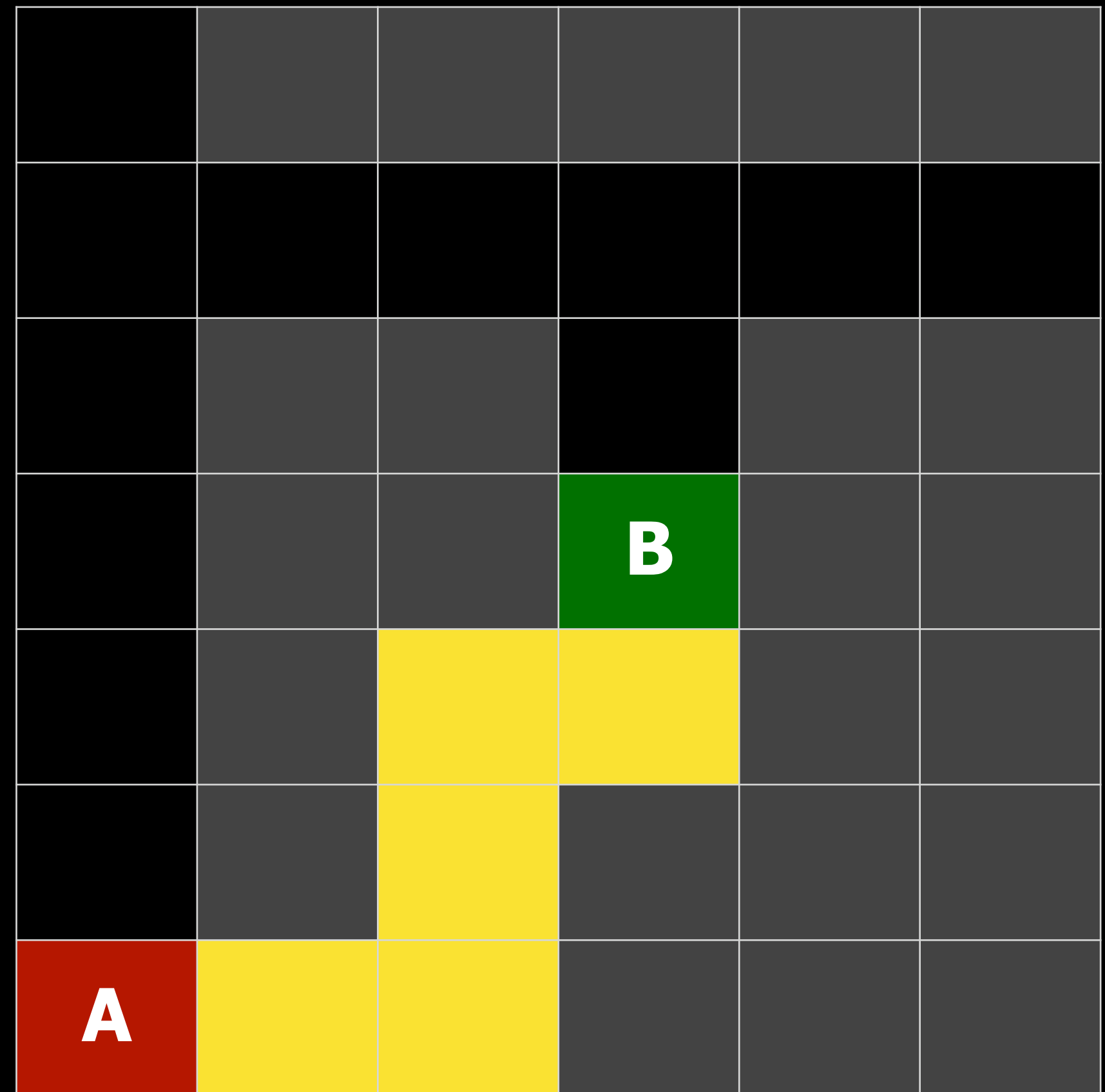
Depth-First Search



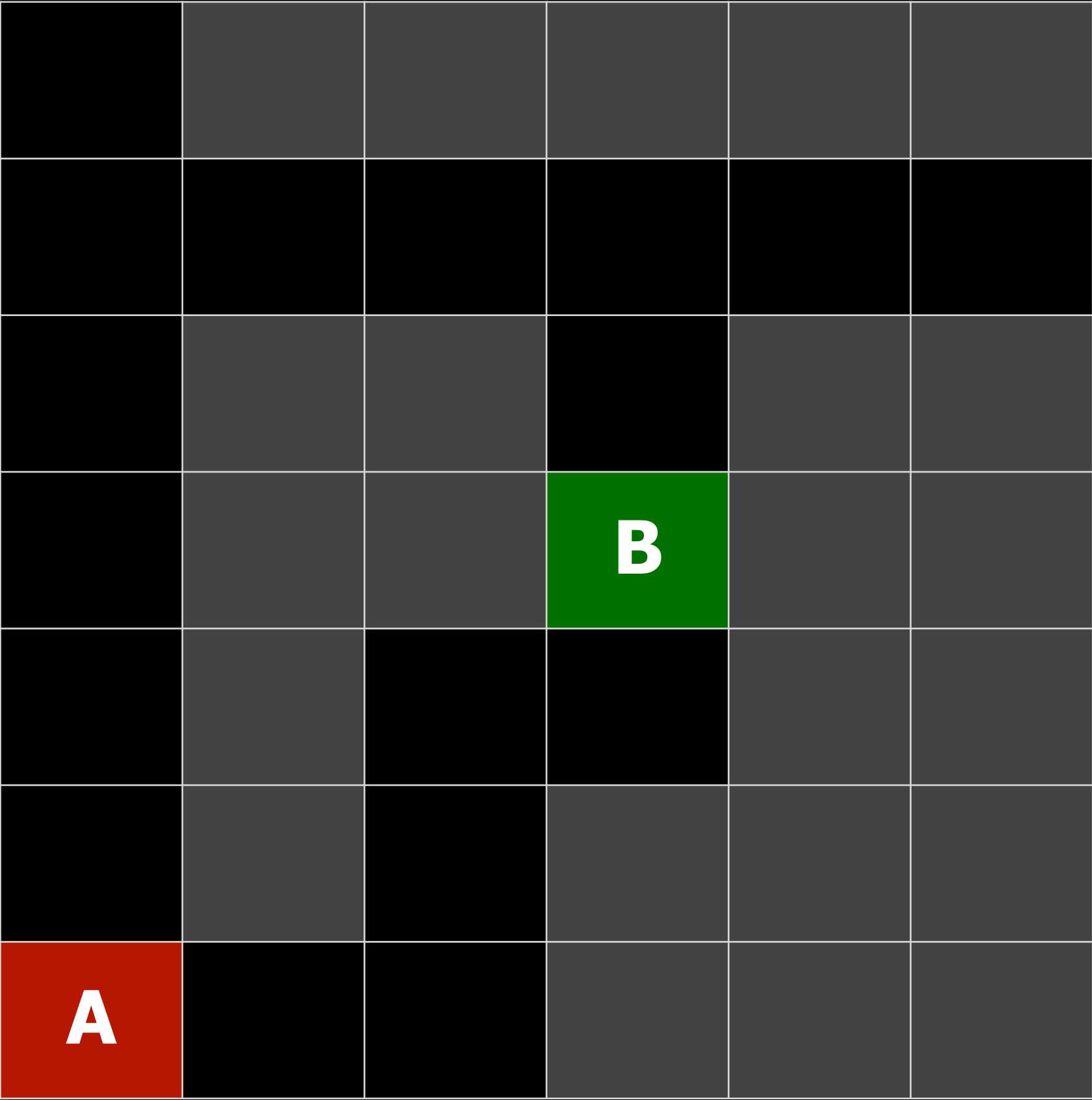
Depth-First Search



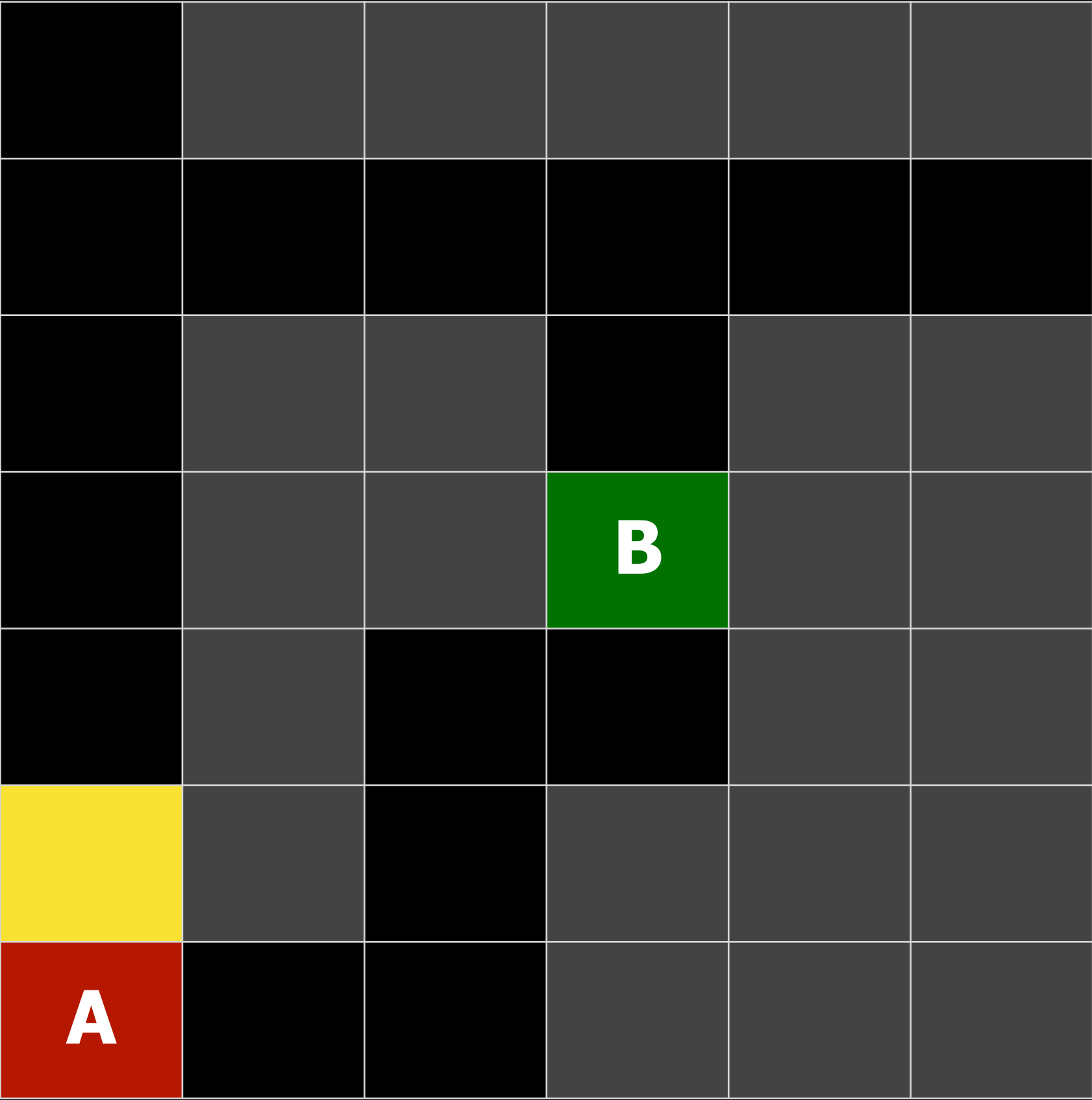
Depth-First Search



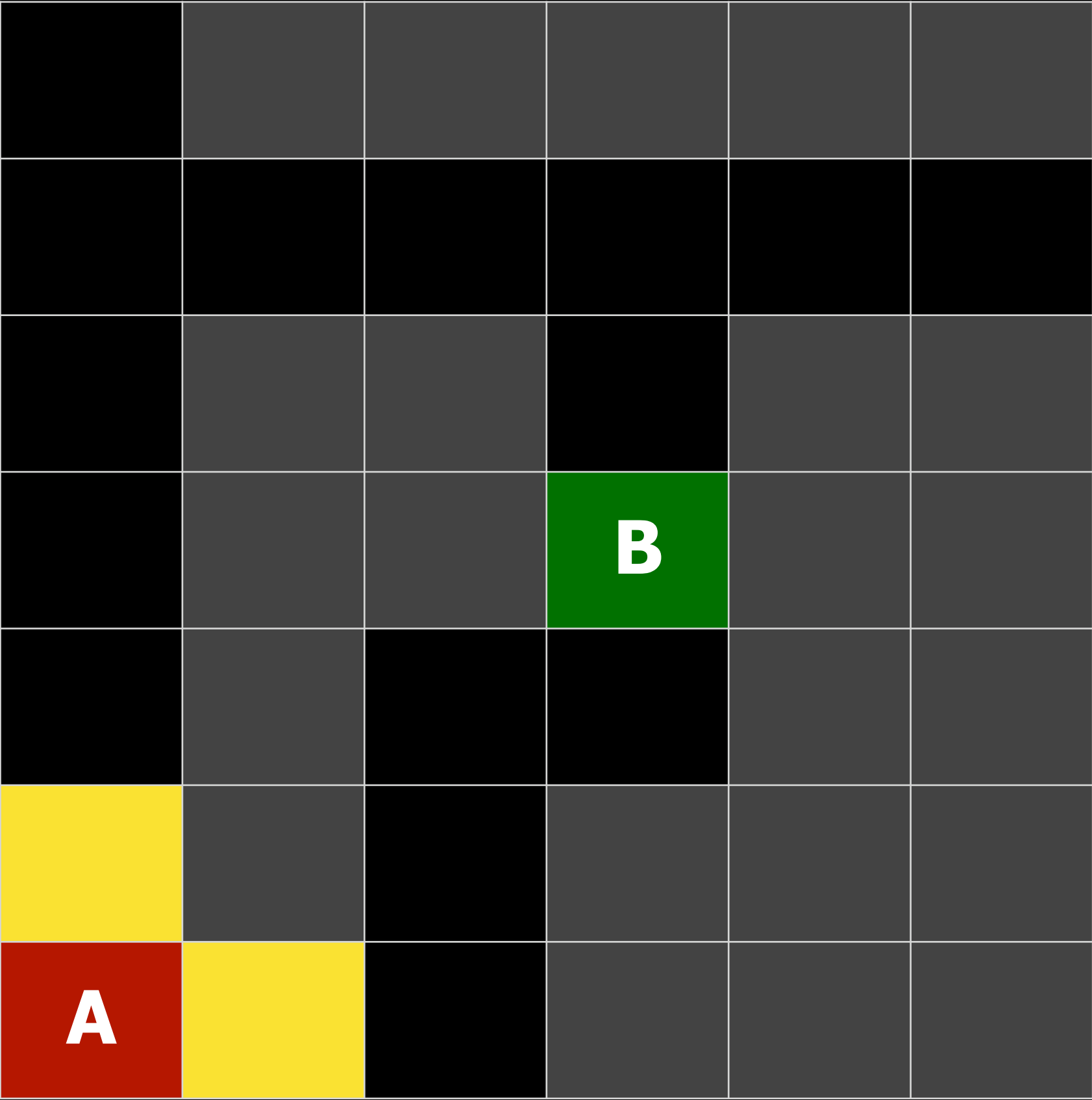
Breadth-First Search



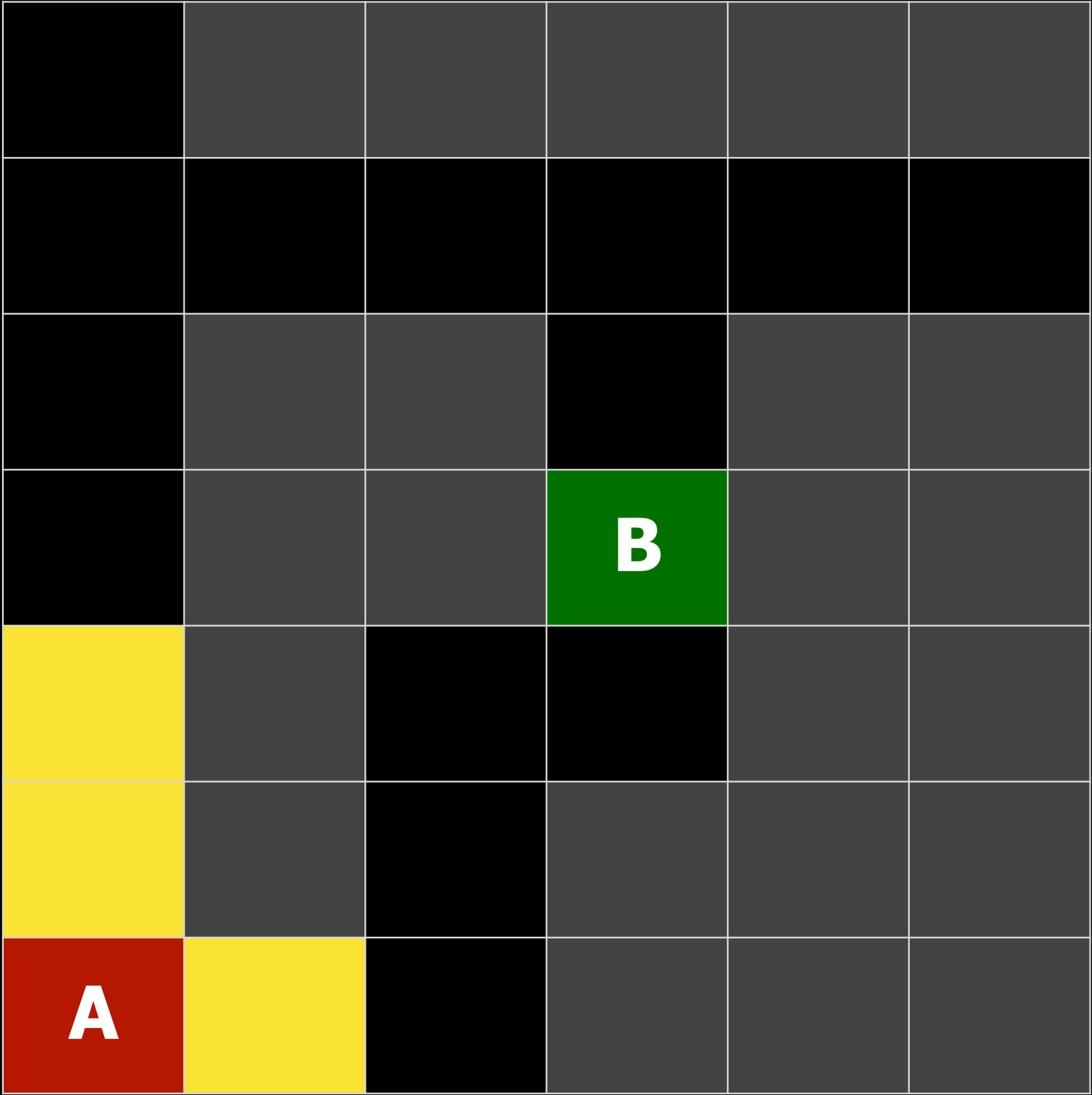
Breadth-First Search



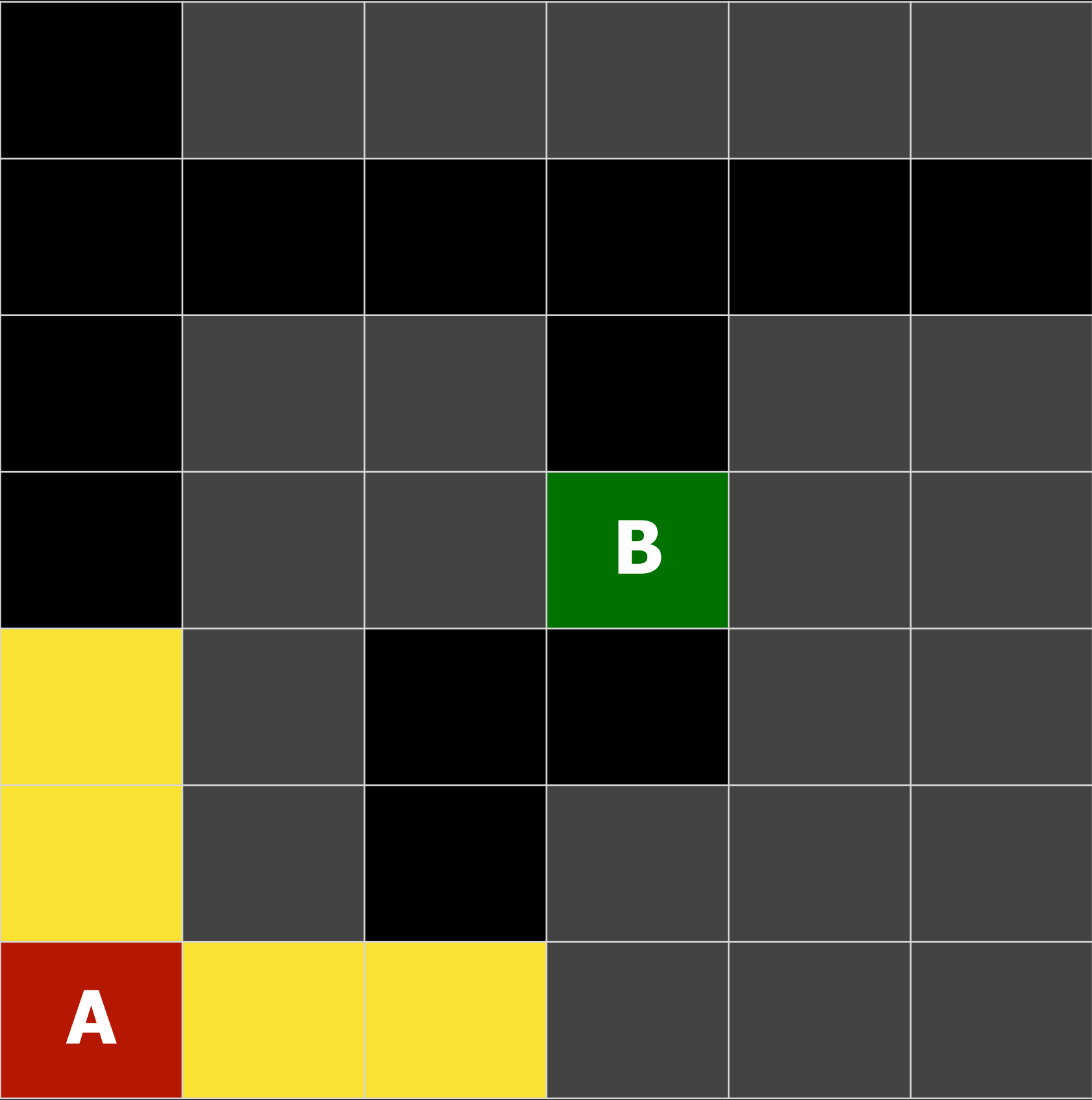
Breadth-First Search



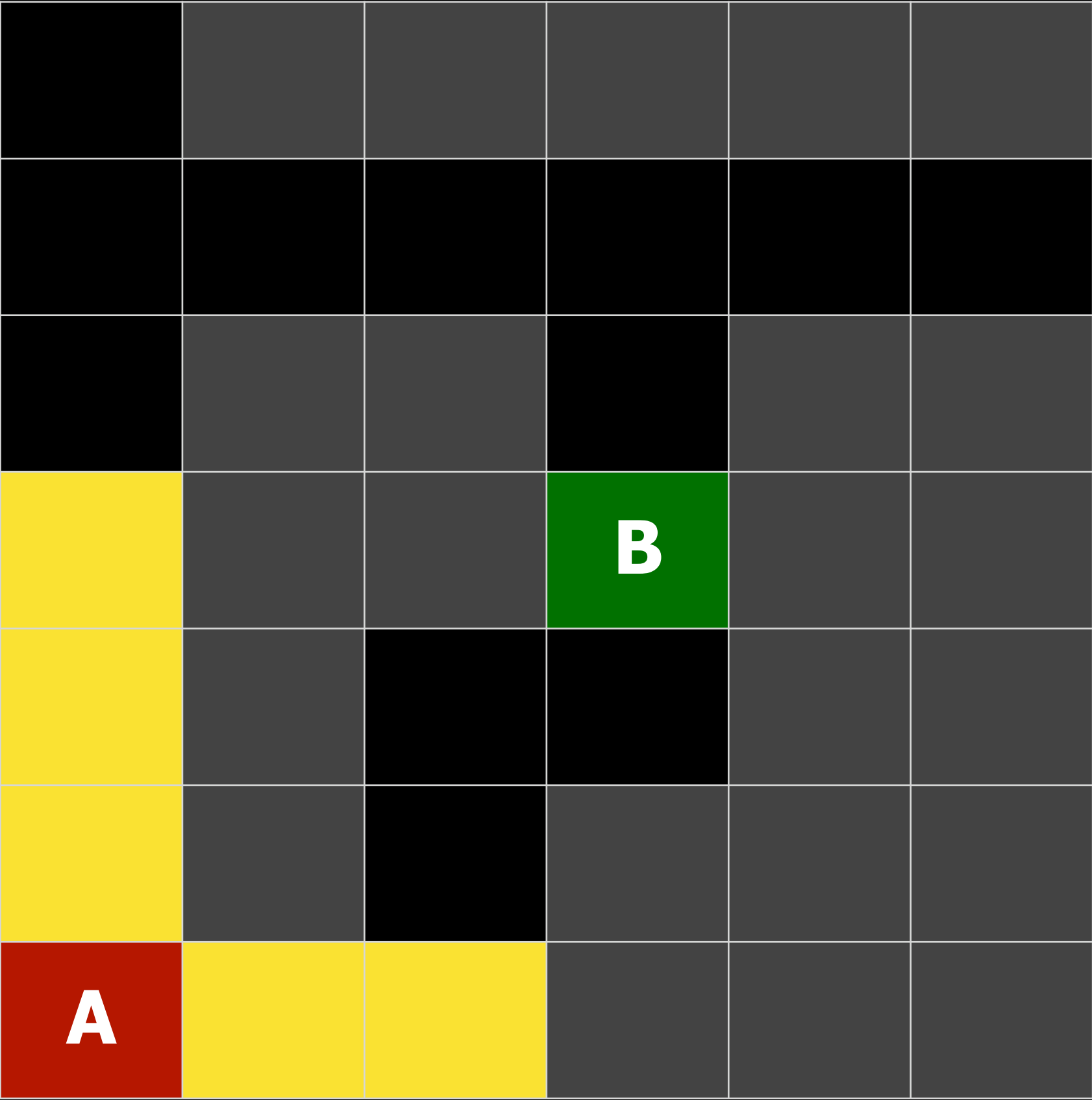
Breadth-First Search



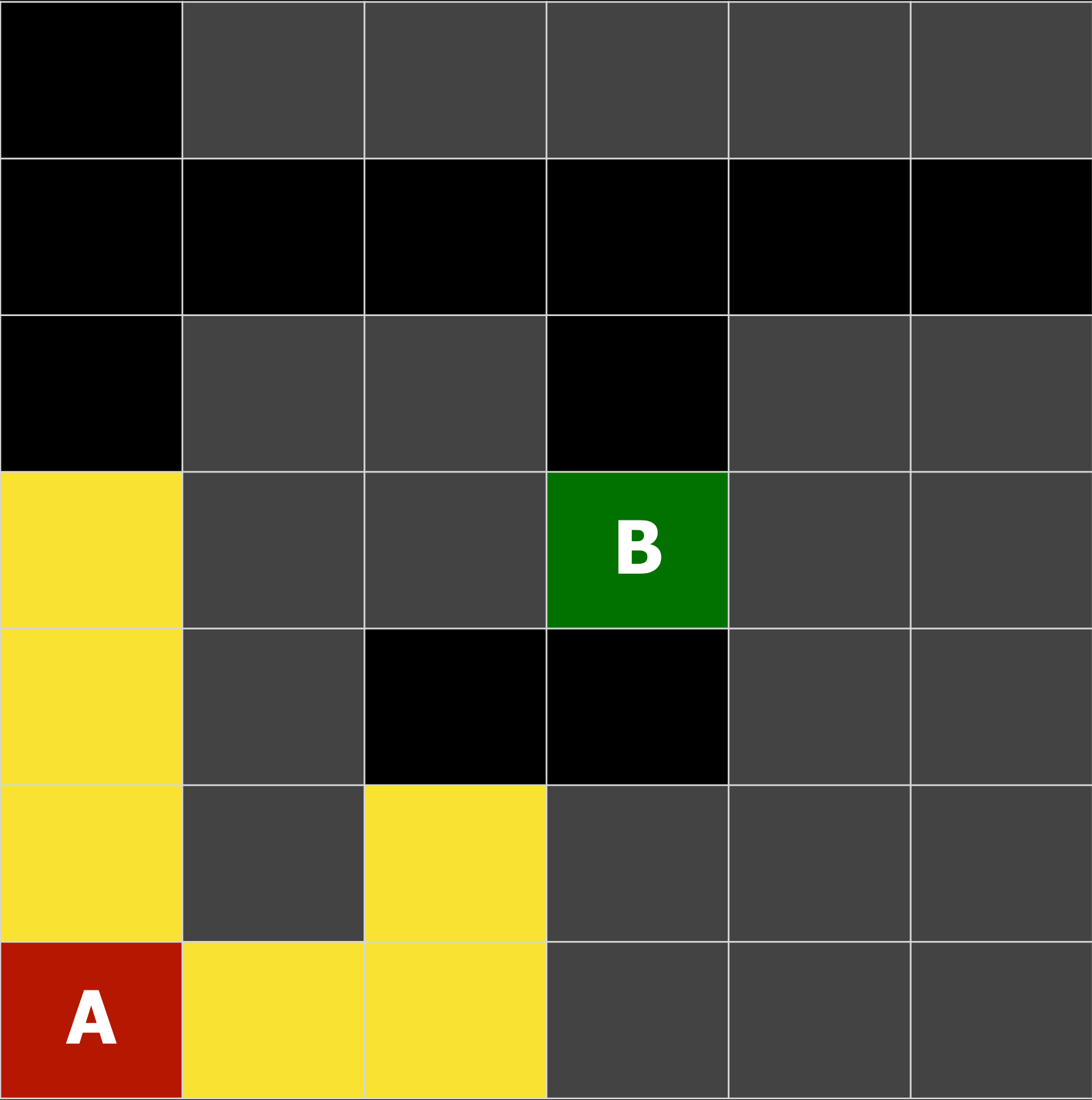
Breadth-First Search



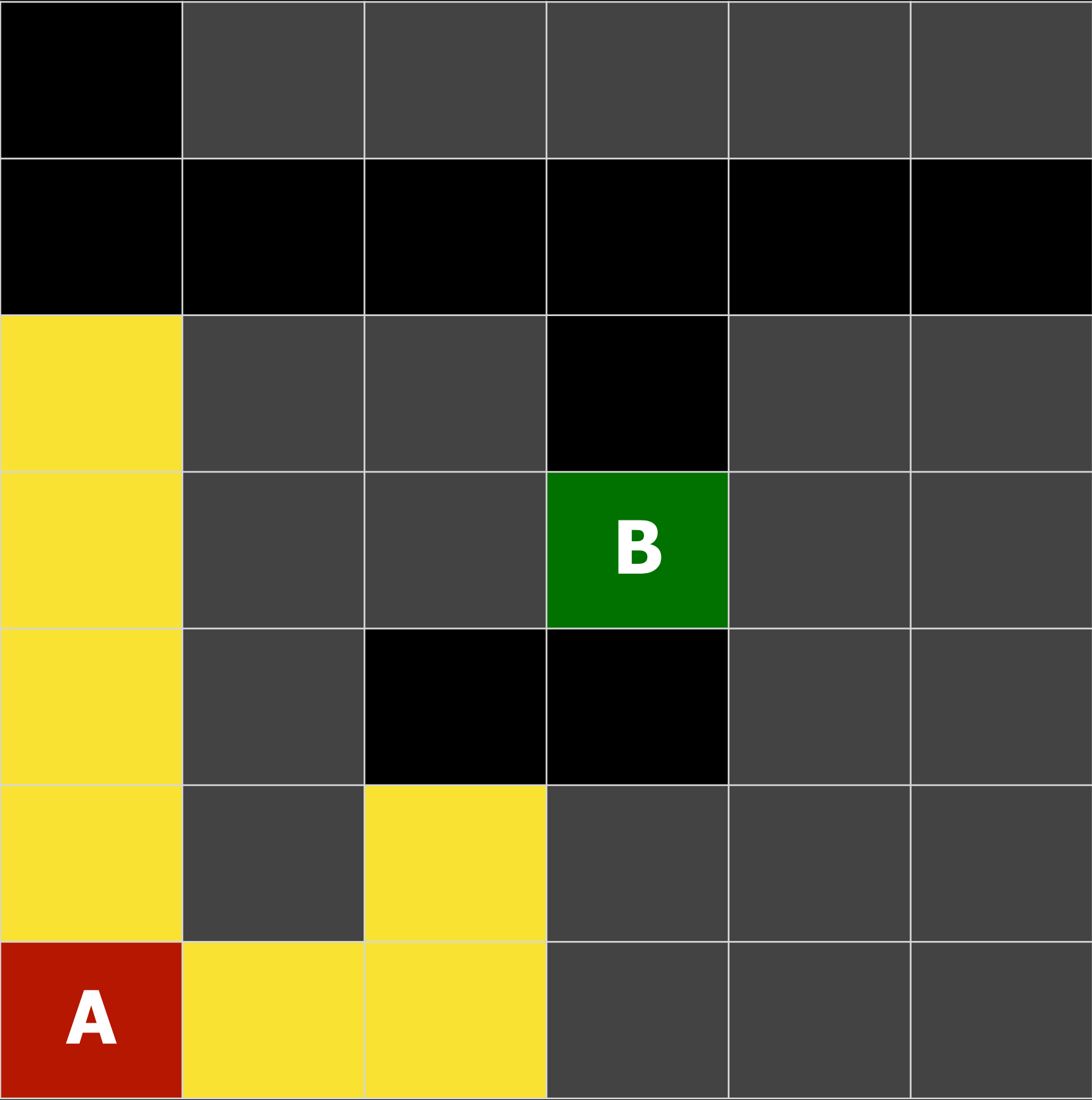
Breadth-First Search



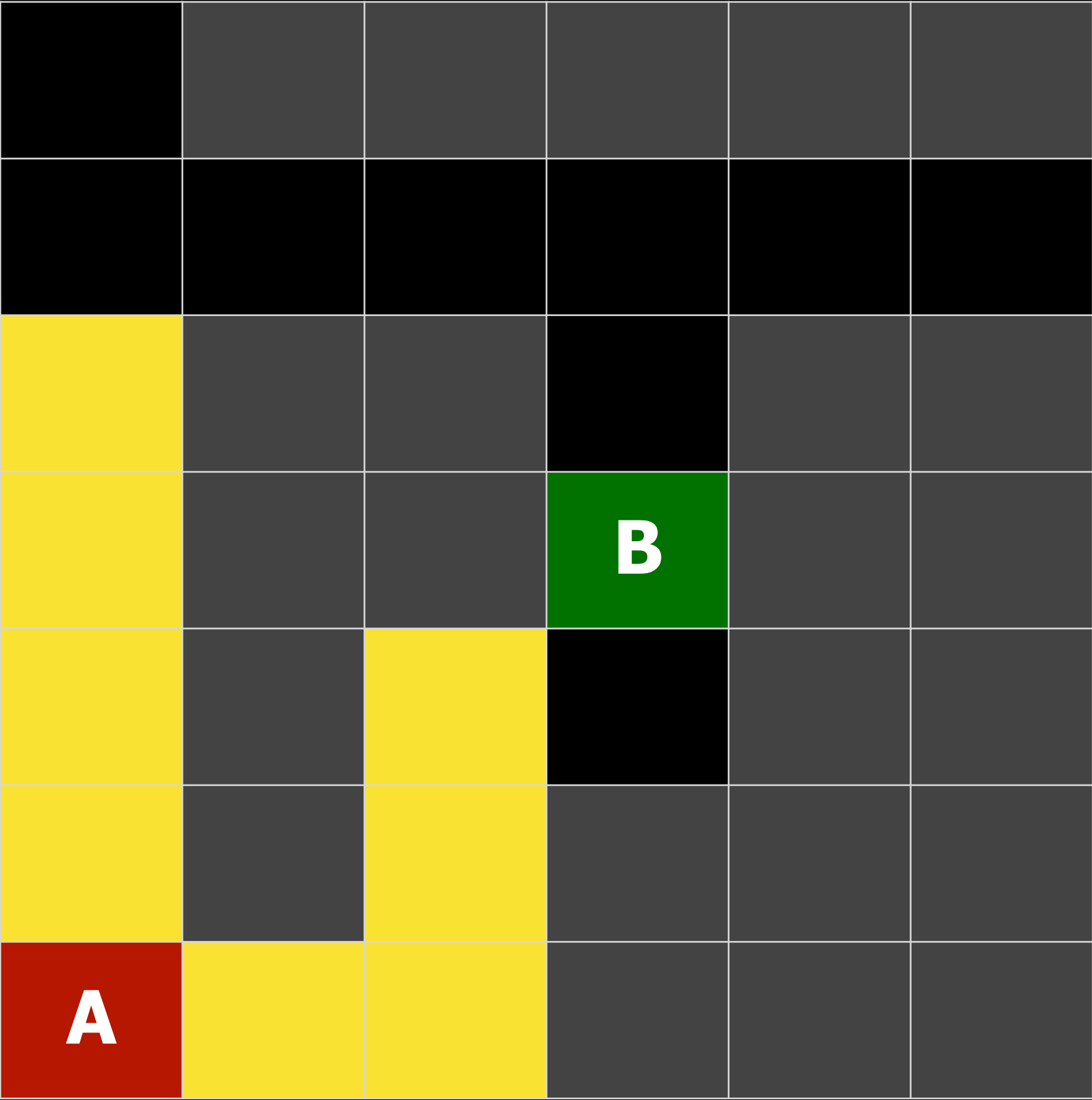
Breadth-First Search



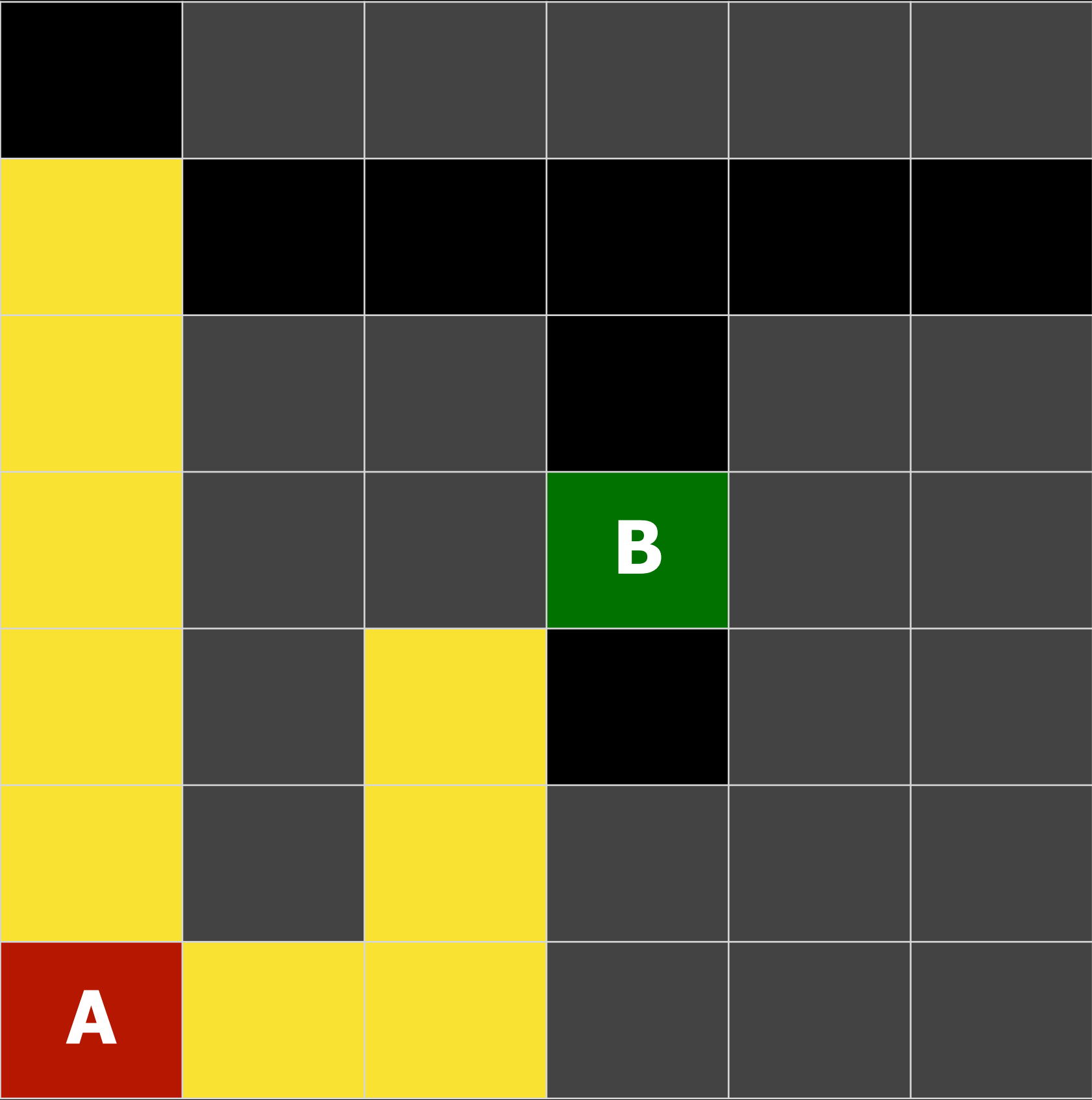
Breadth-First Search



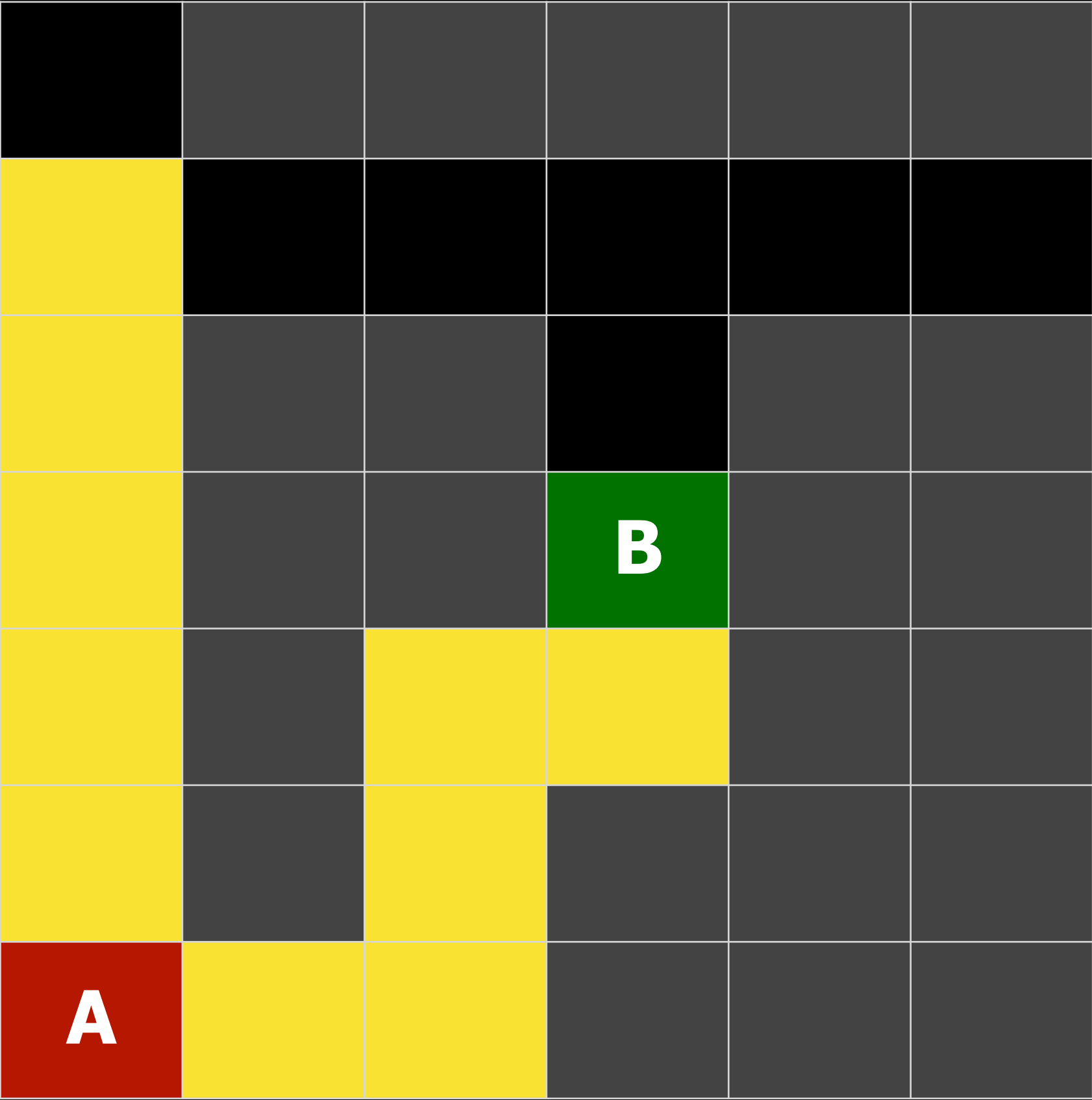
Breadth-First Search



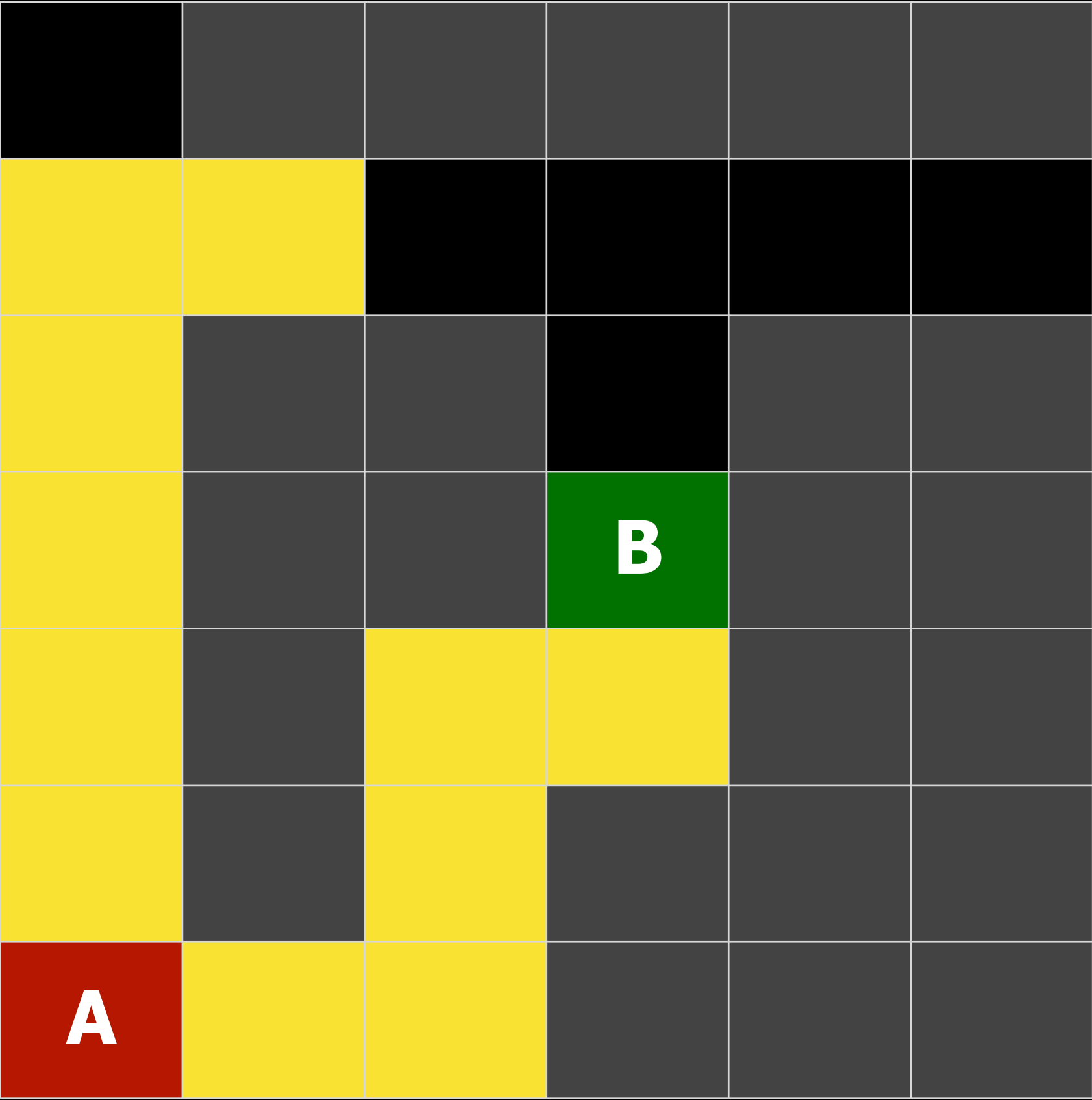
Breadth-First Search



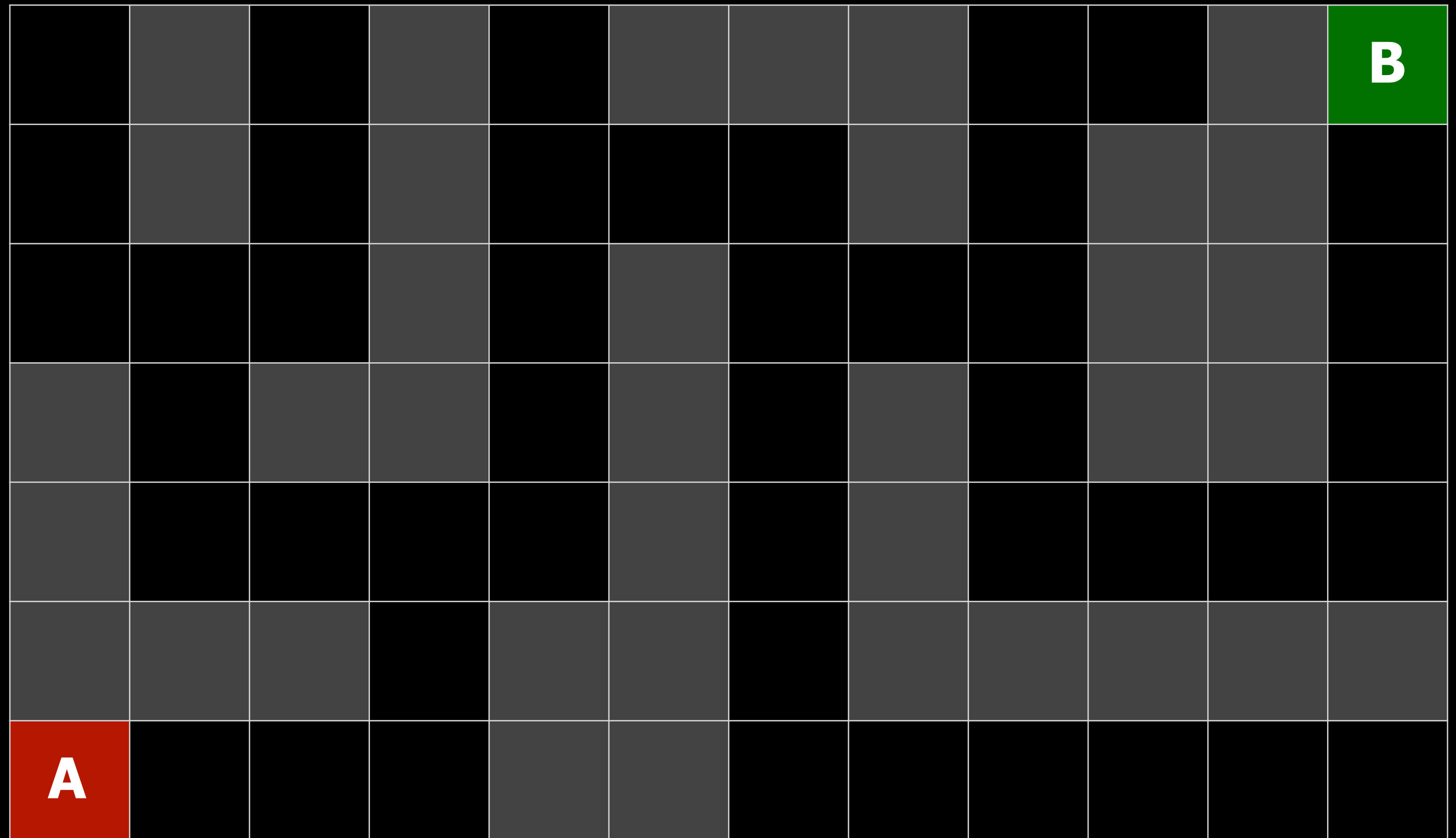
Breadth-First Search



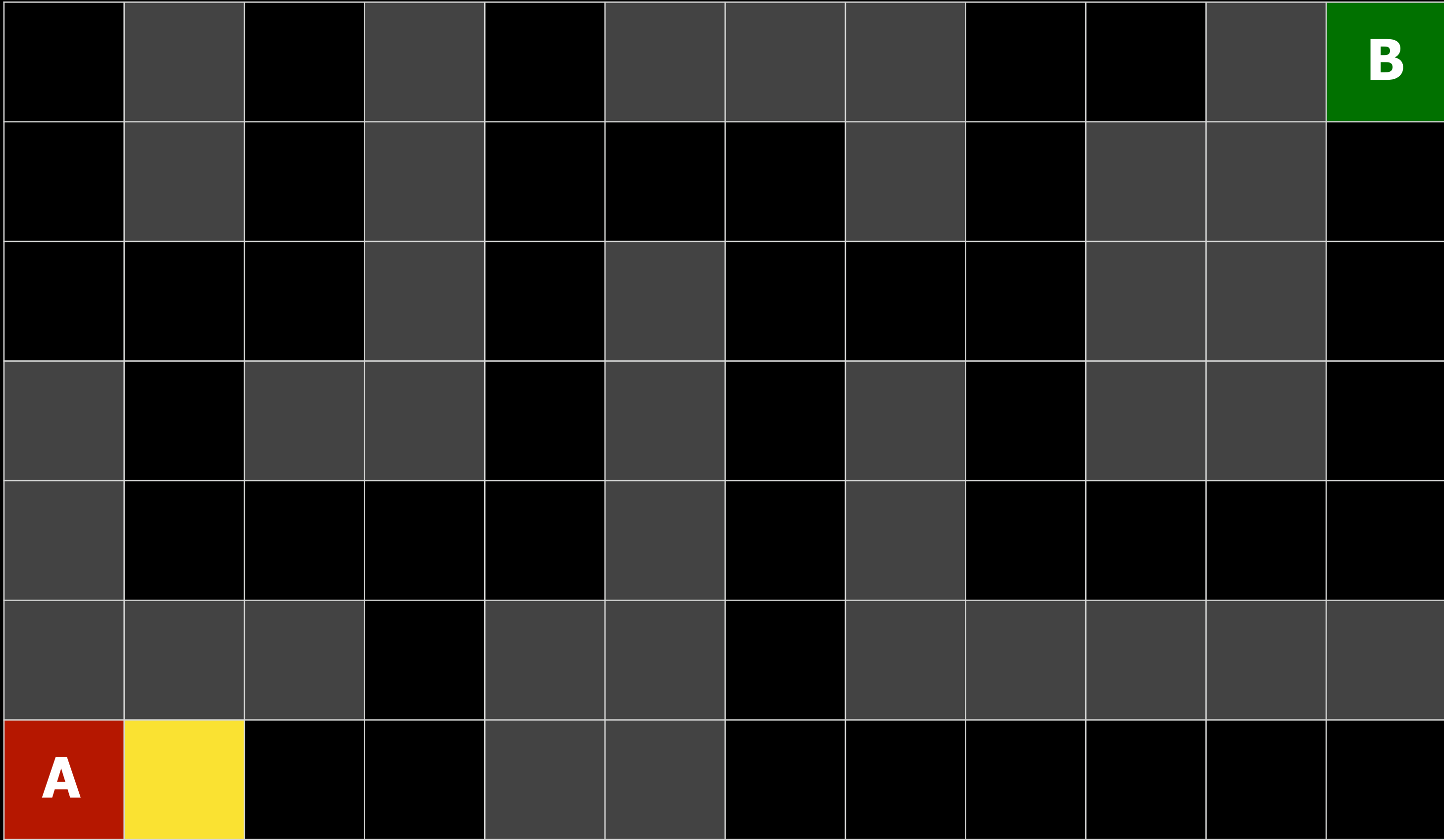
Breadth-First Search



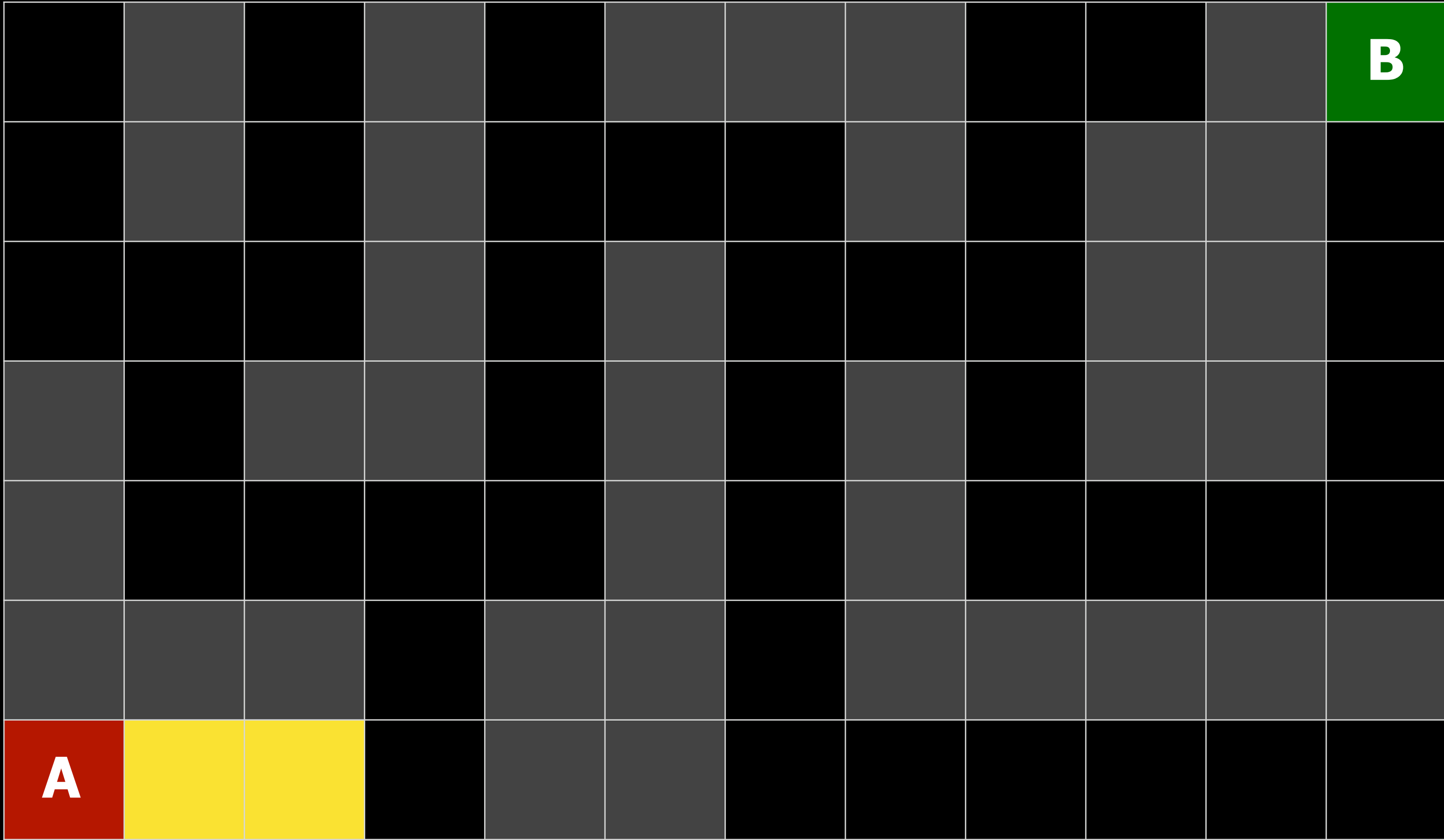
Breadth-First Search



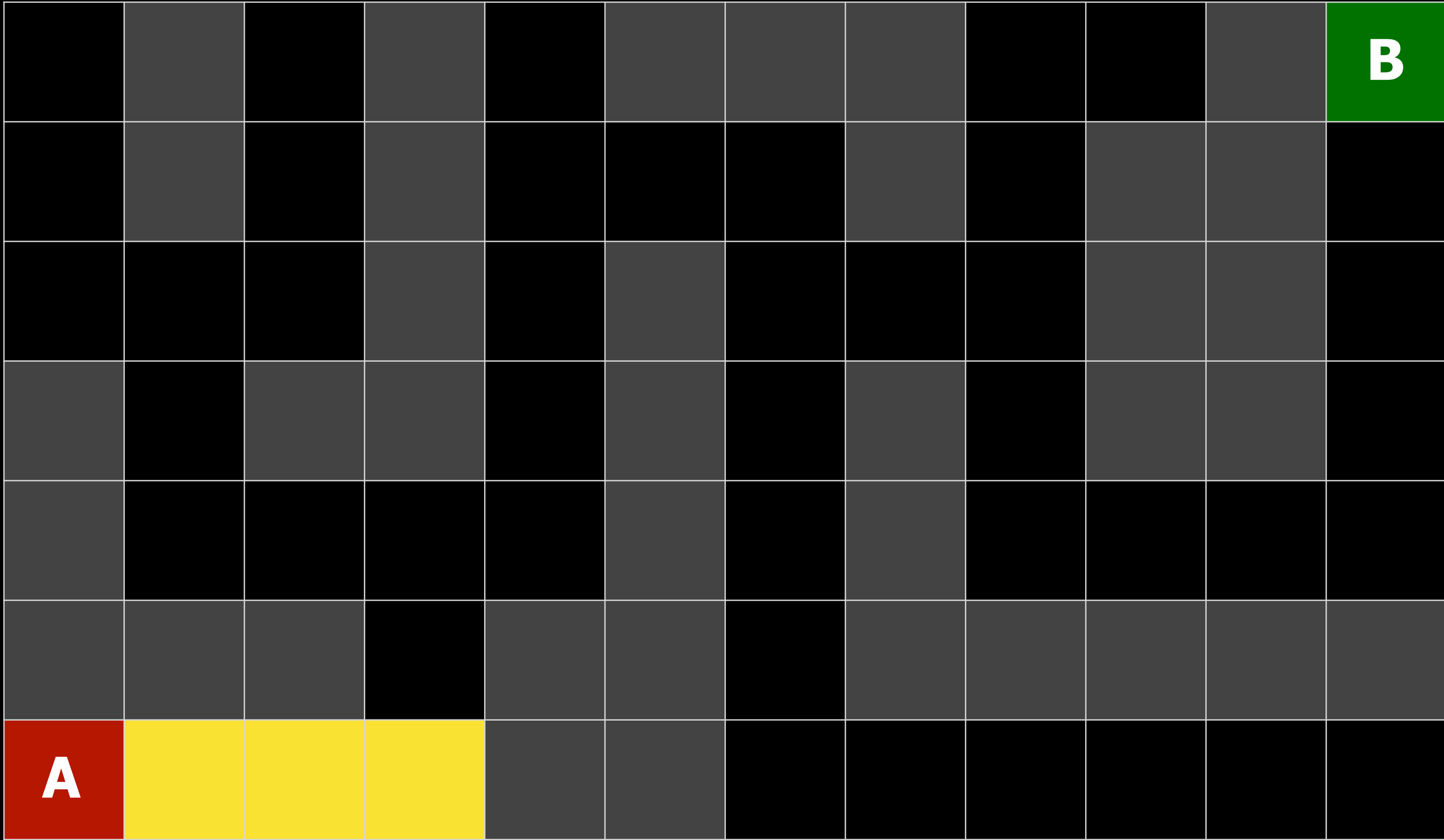
Breadth-First Search



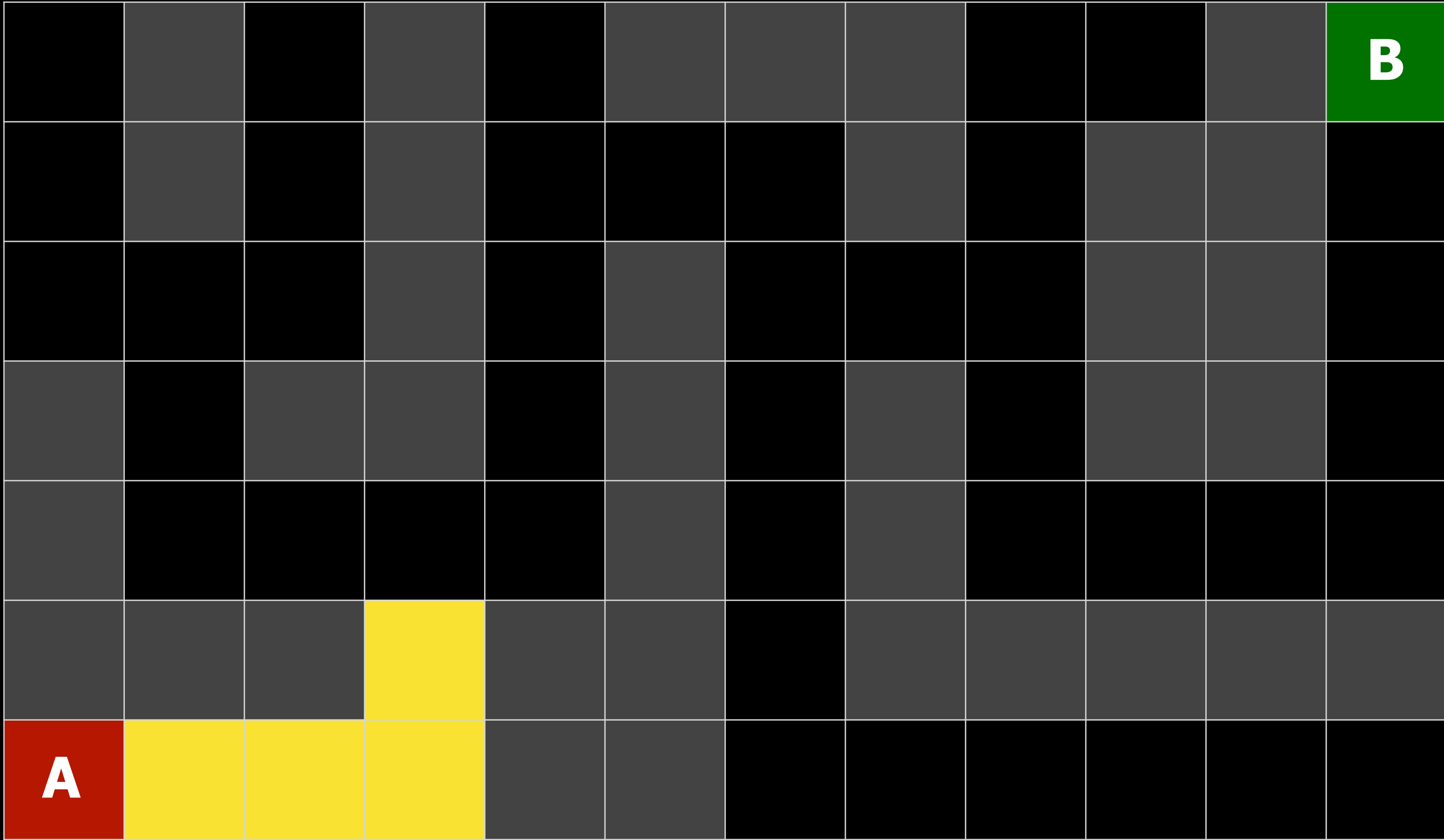
Breadth-First Search



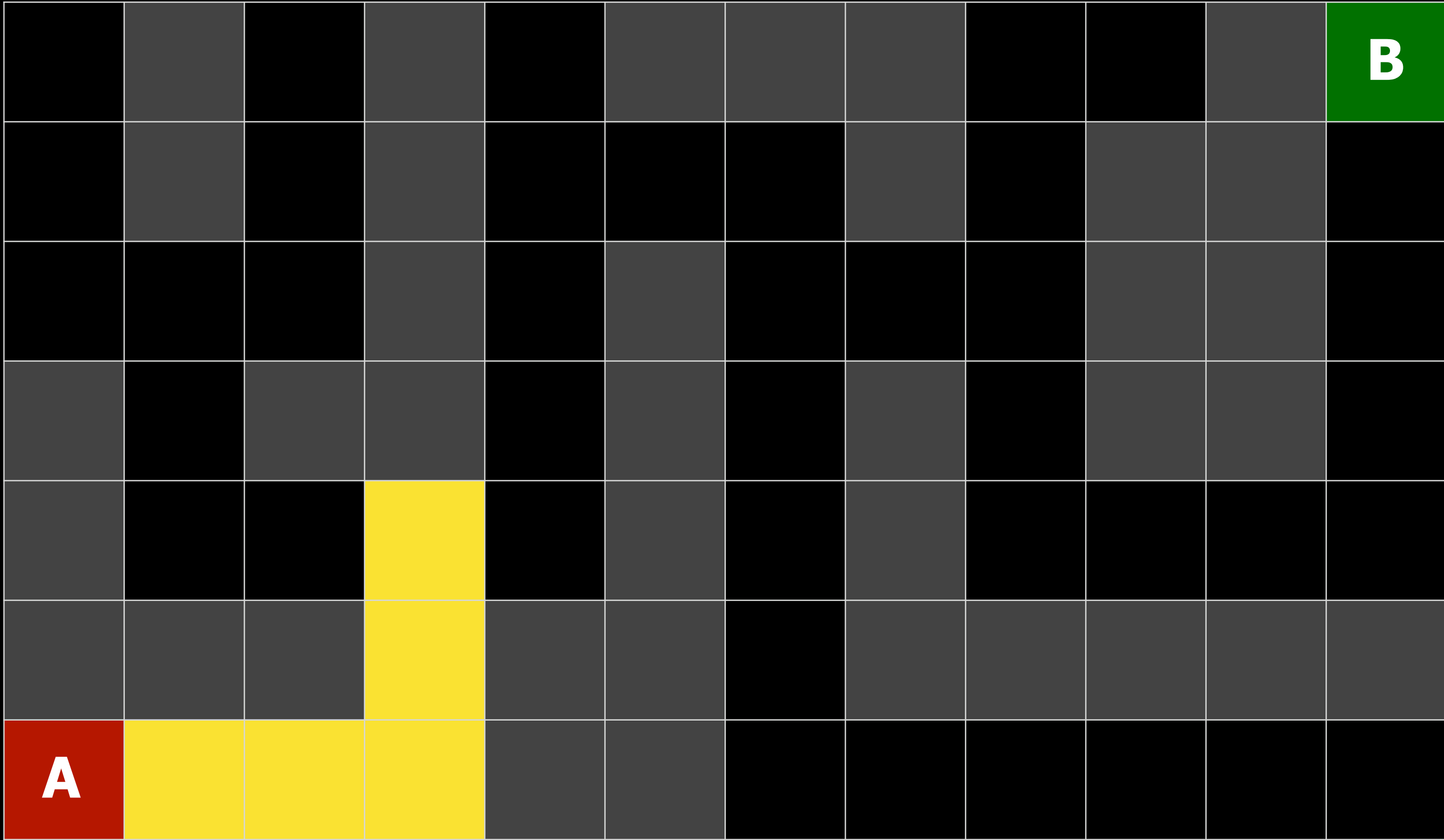
Breadth-First Search



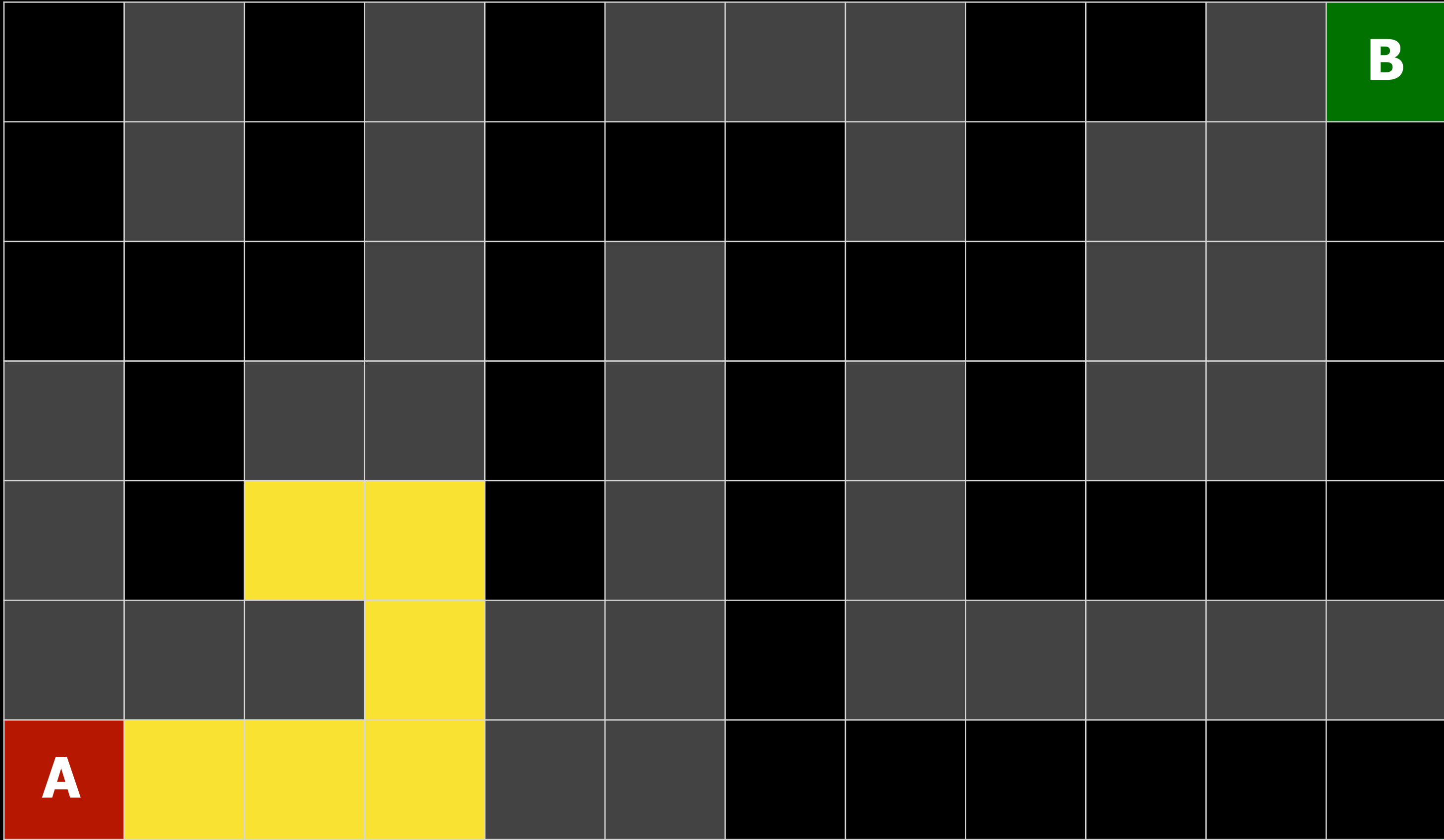
Breadth-First Search



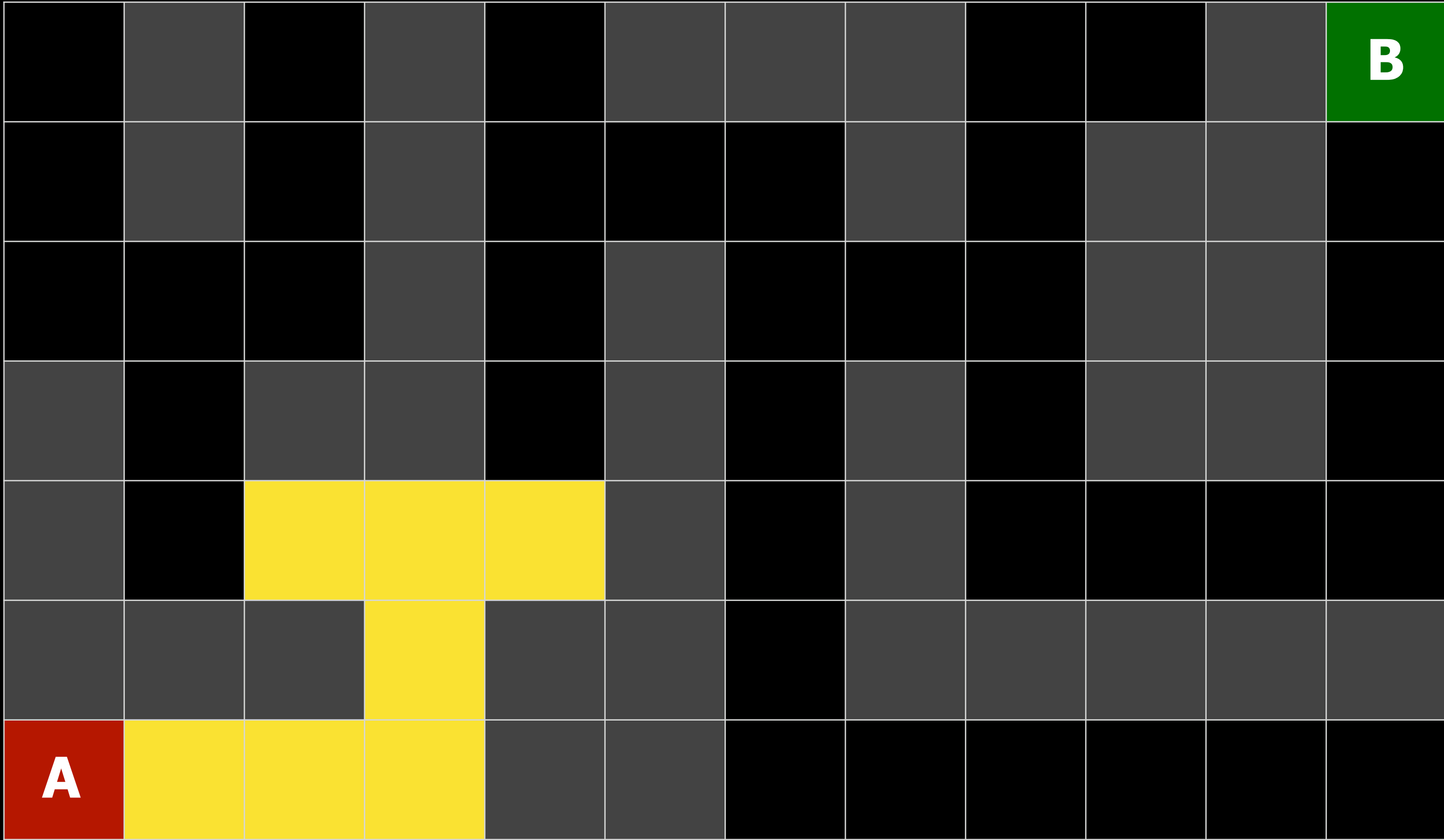
Breadth-First Search



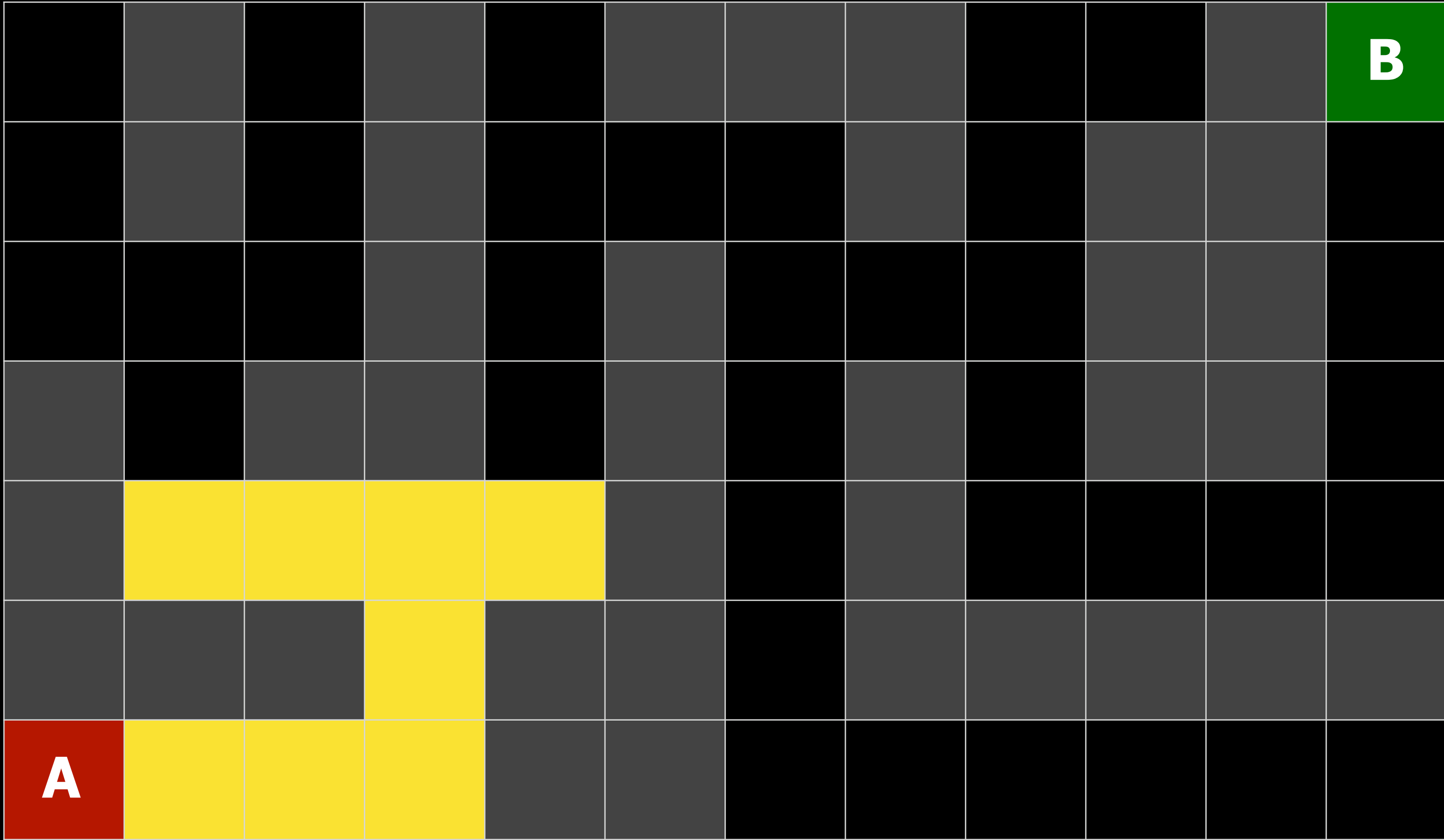
Breadth-First Search



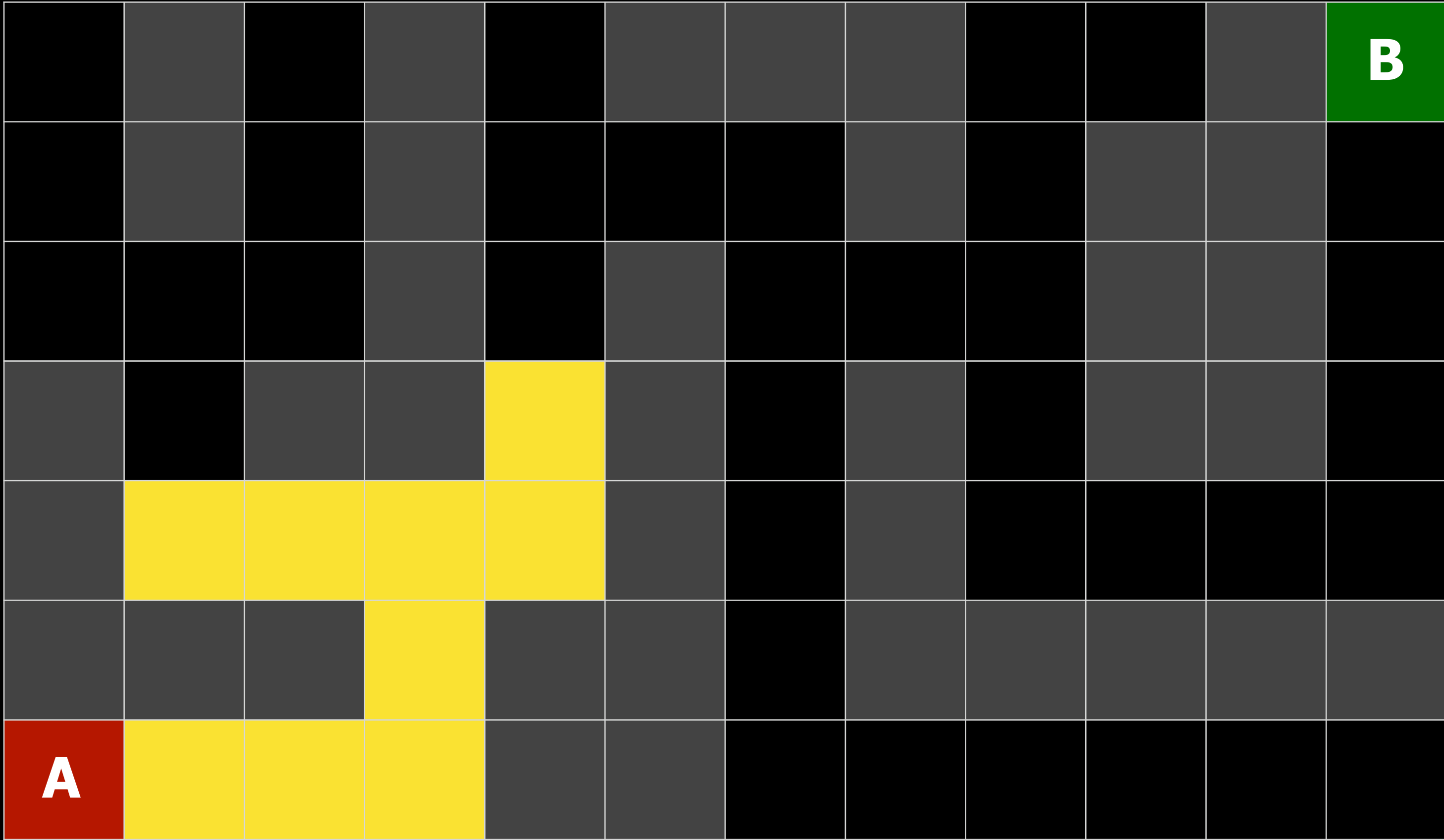
Breadth-First Search



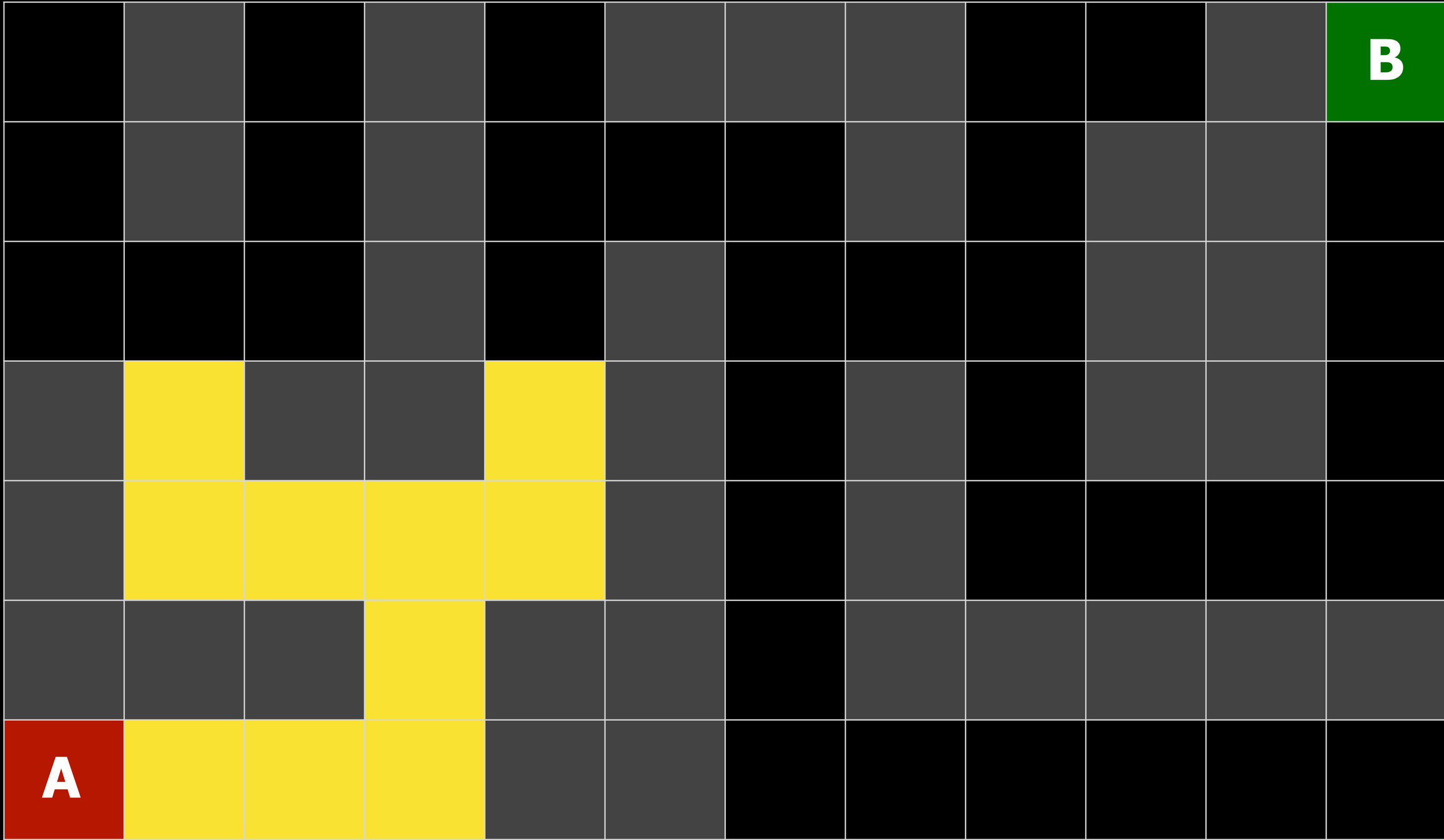
Breadth-First Search



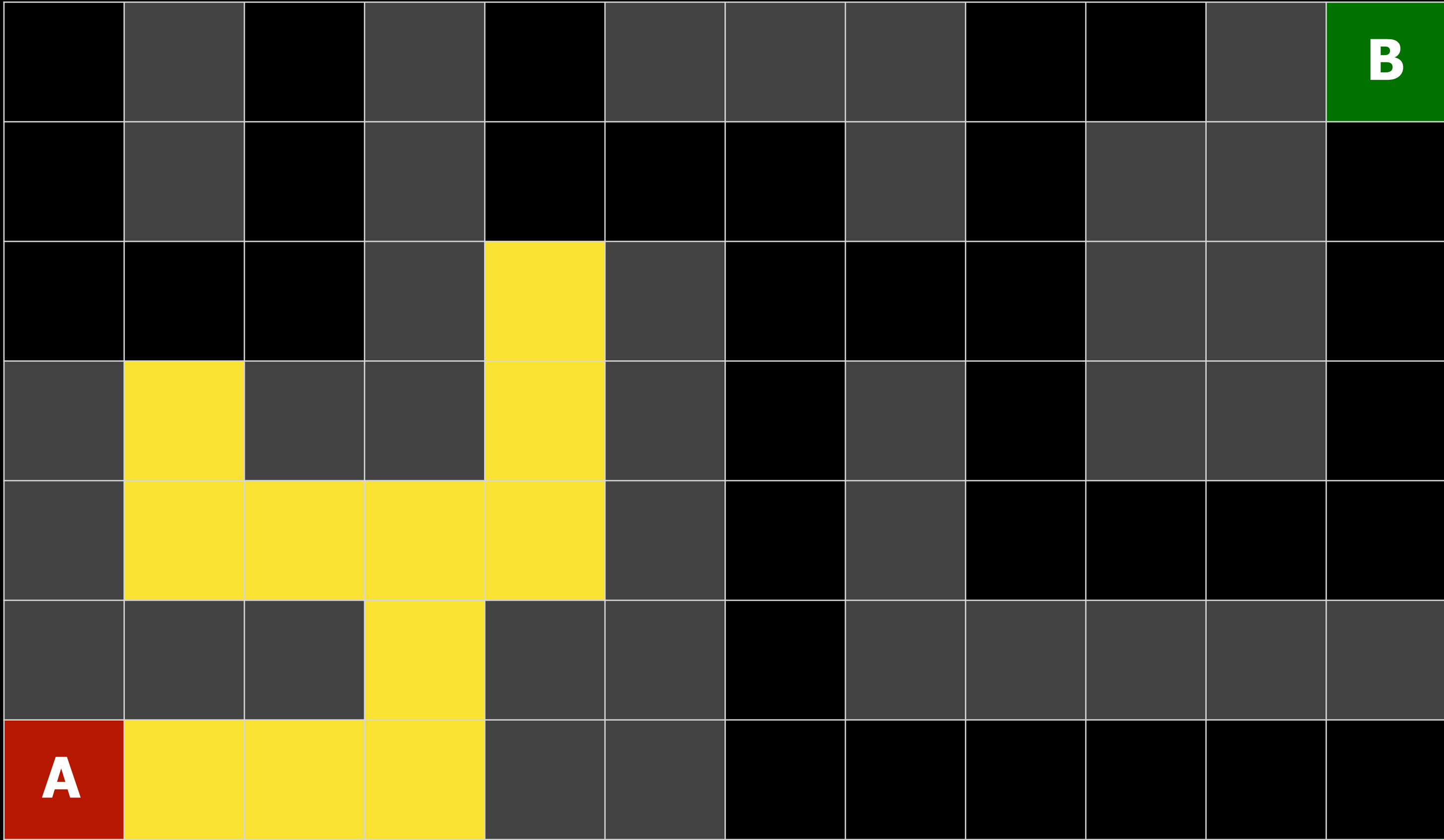
Breadth-First Search



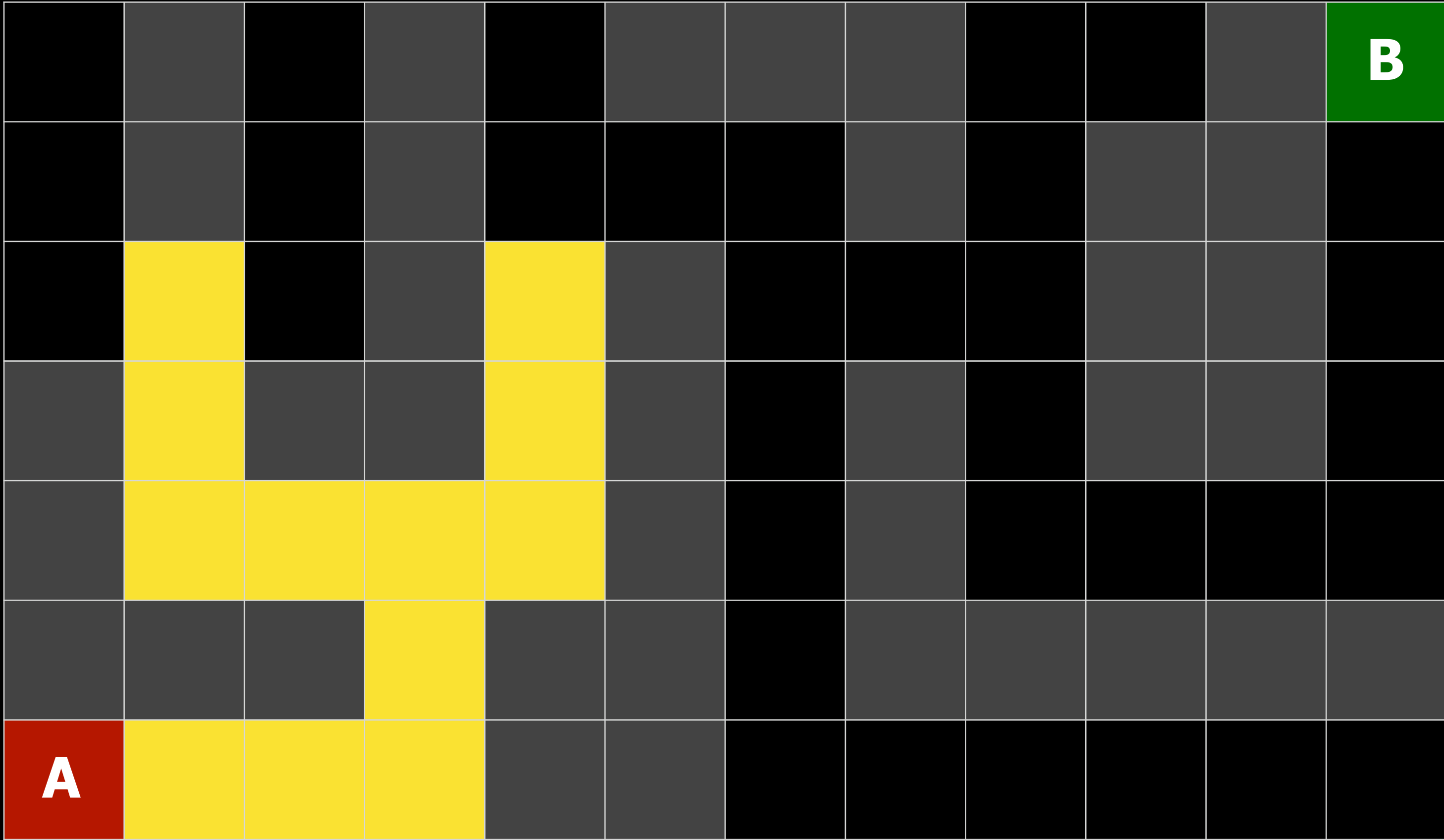
Breadth-First Search



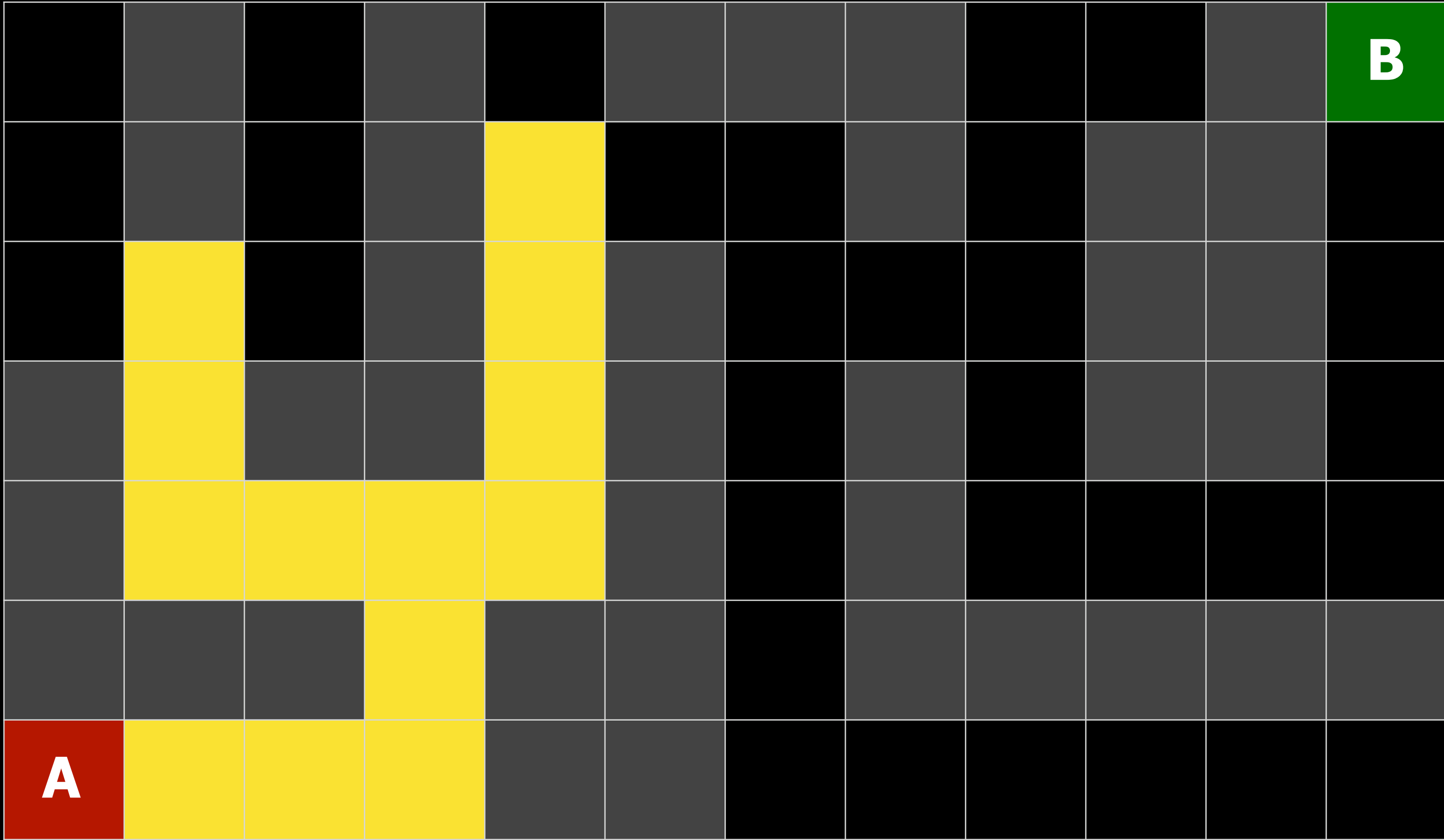
Breadth-First Search



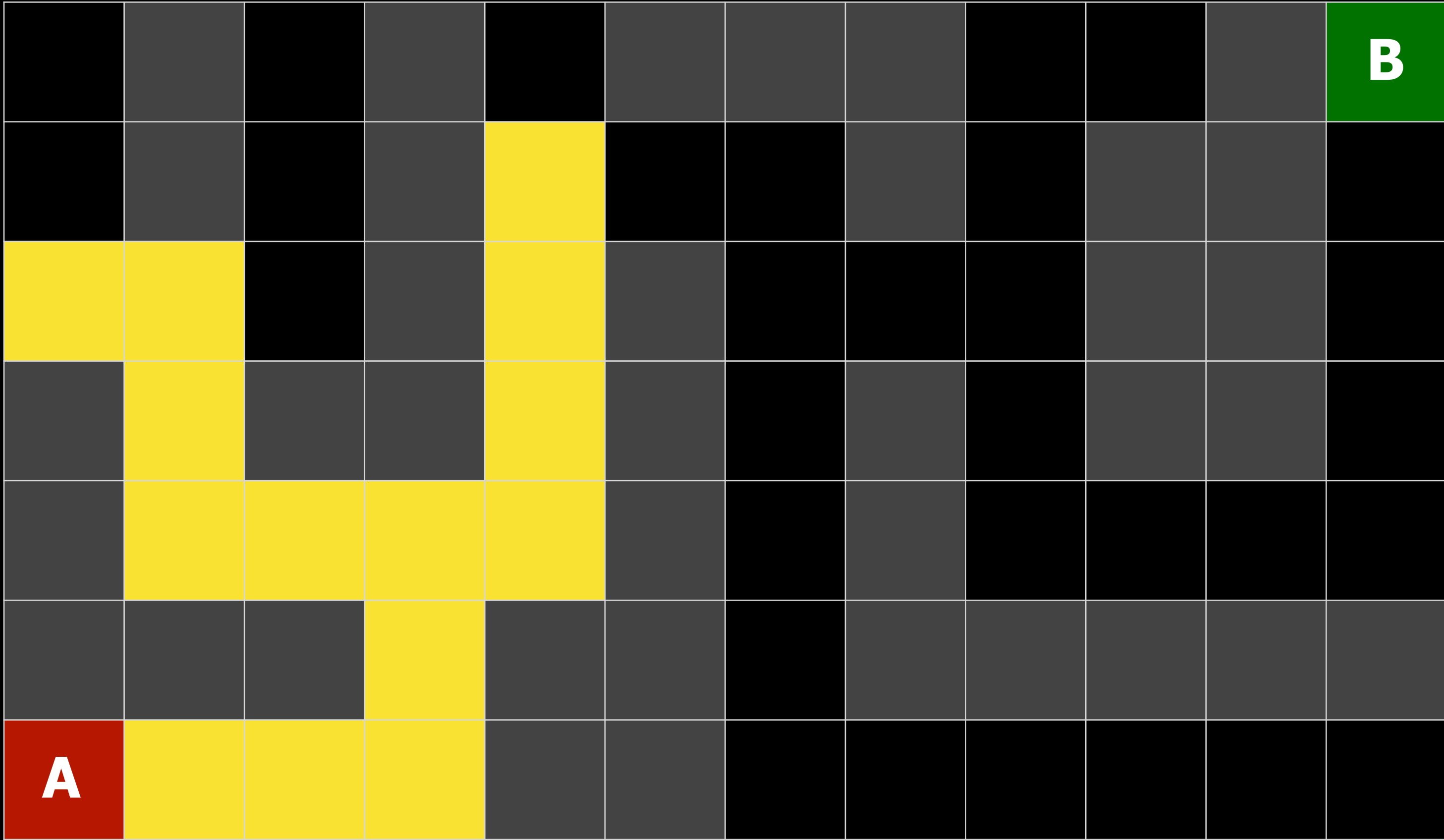
Breadth-First Search



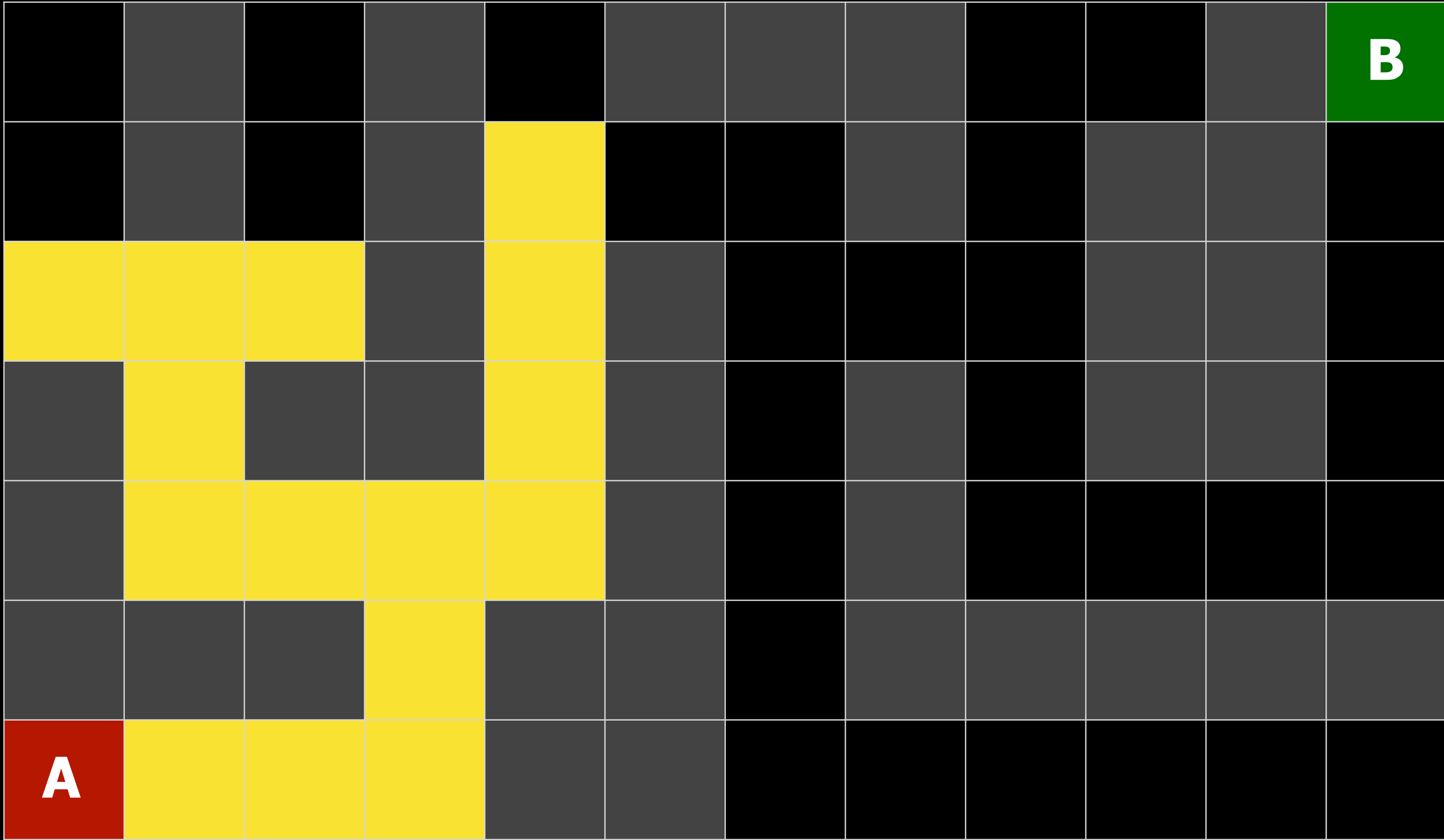
Breadth-First Search



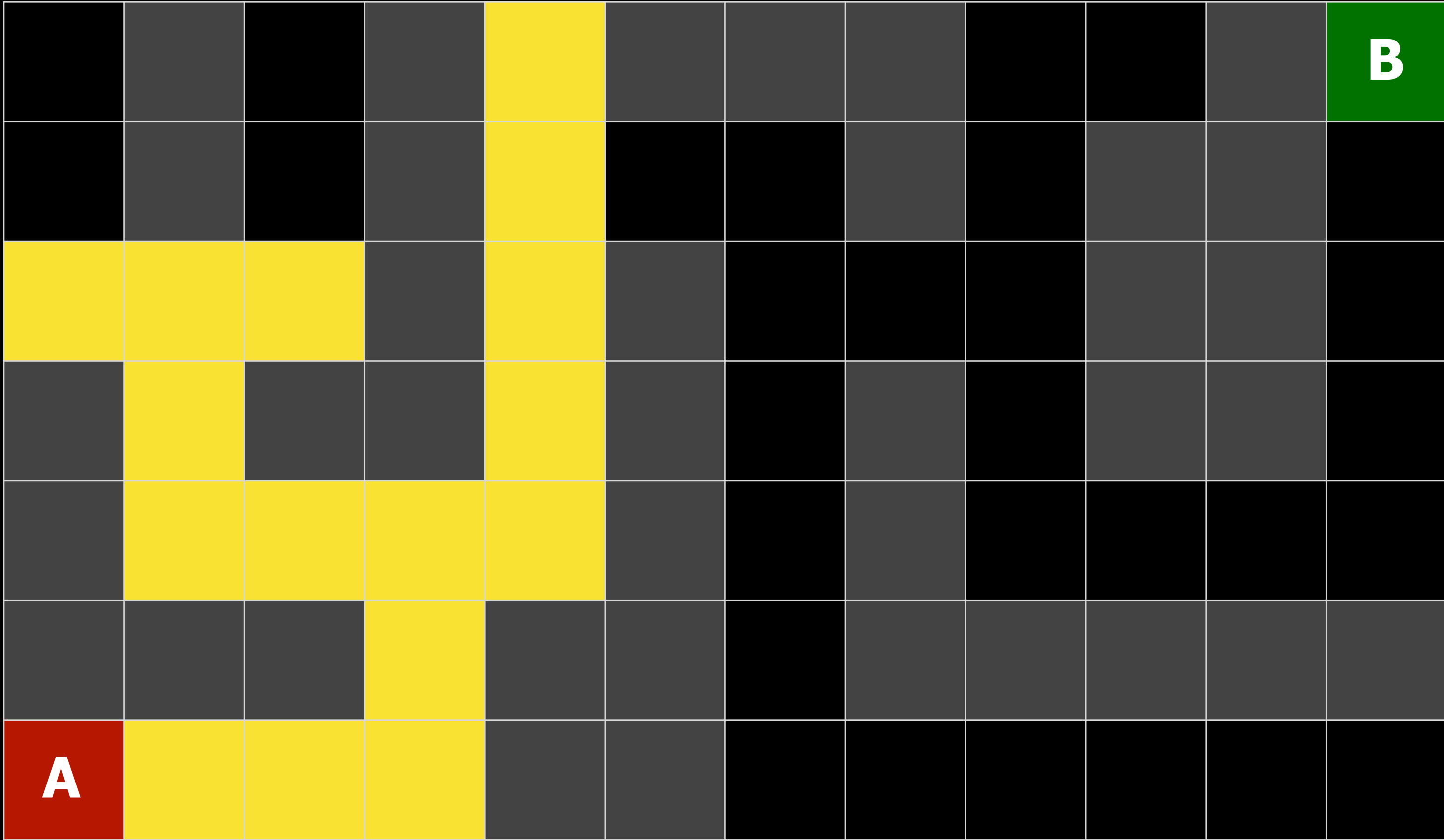
Breadth-First Search



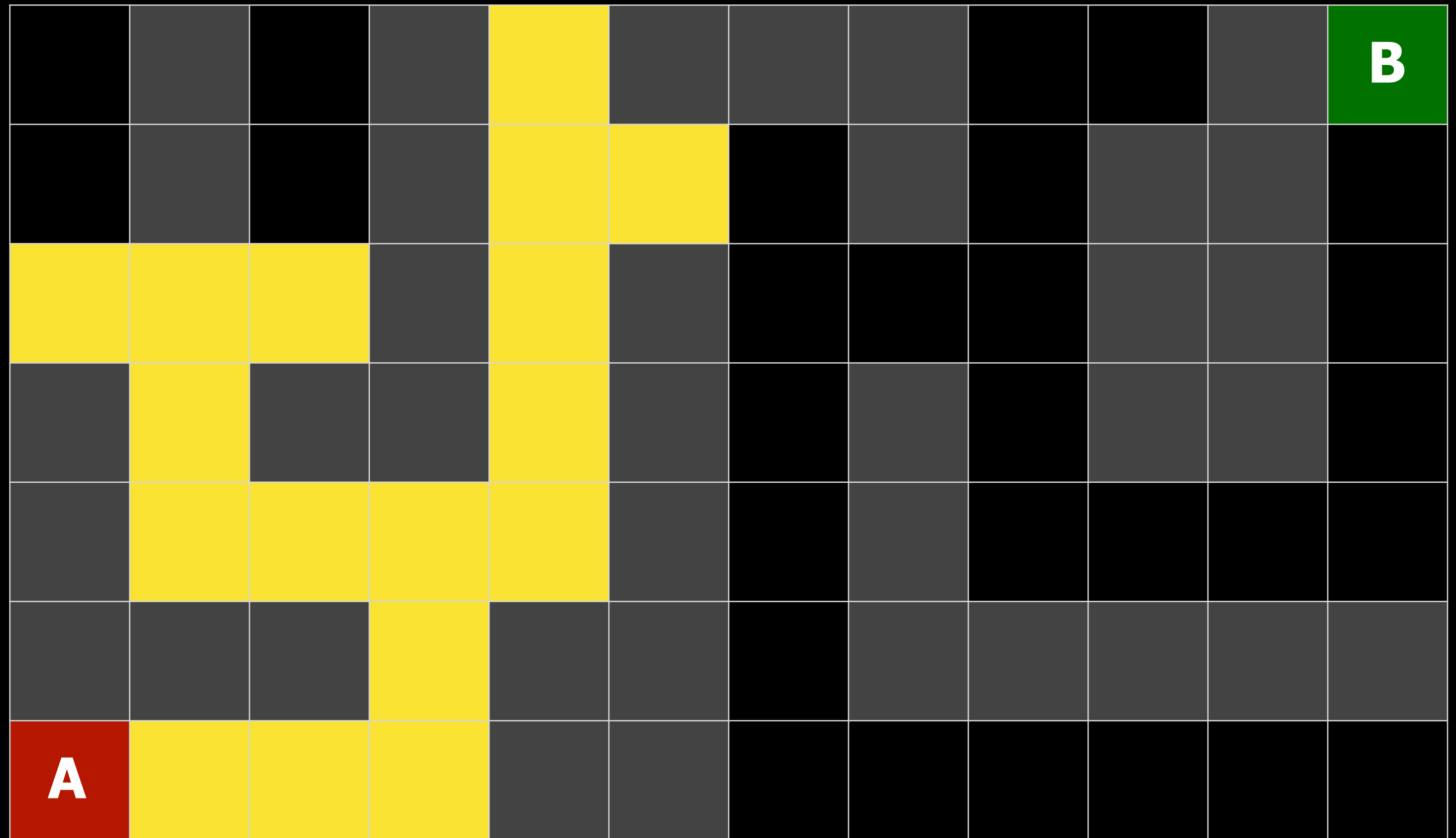
Breadth-First Search



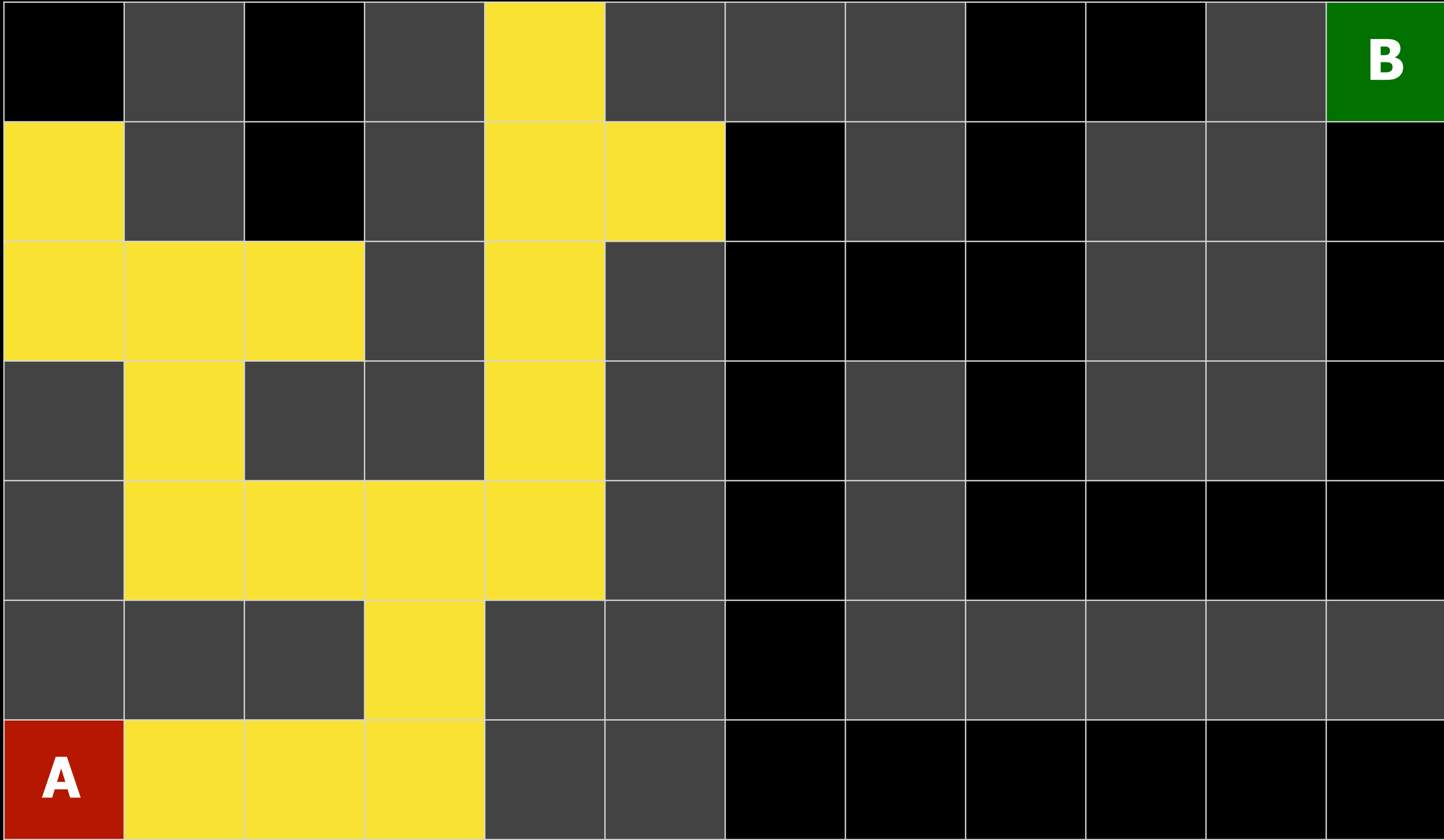
Breadth-First Search



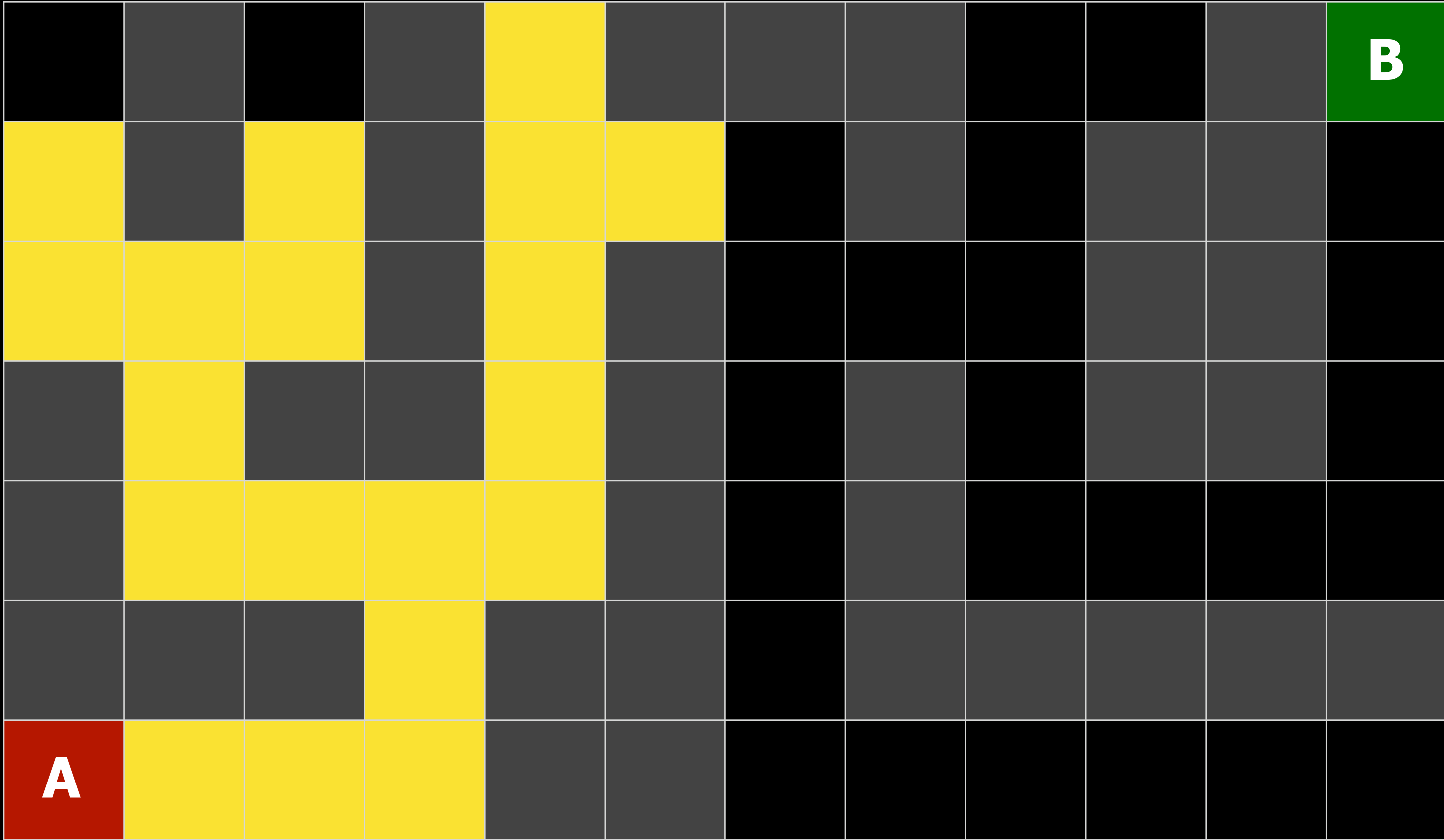
Breadth-First Search



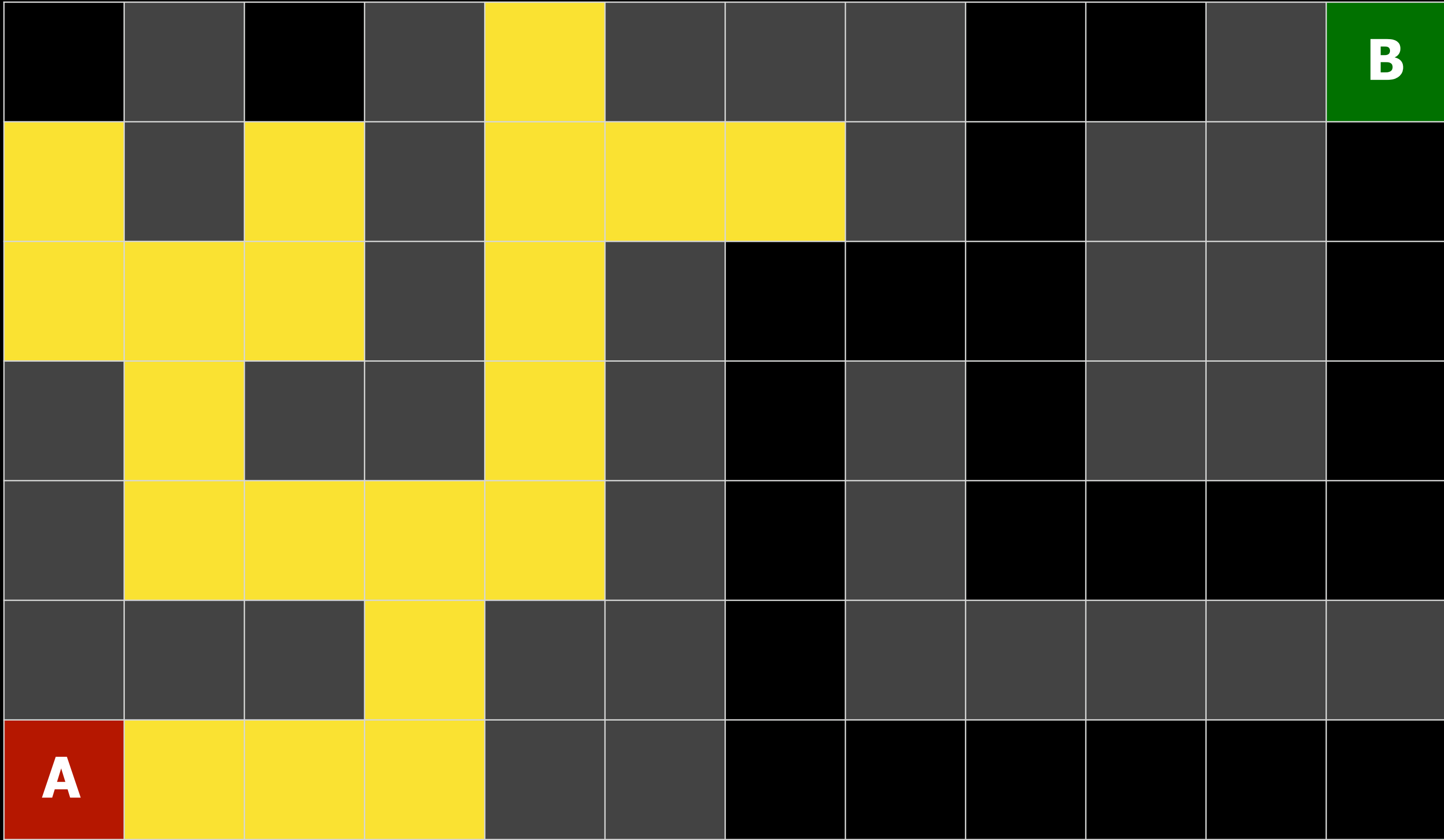
Breadth-First Search



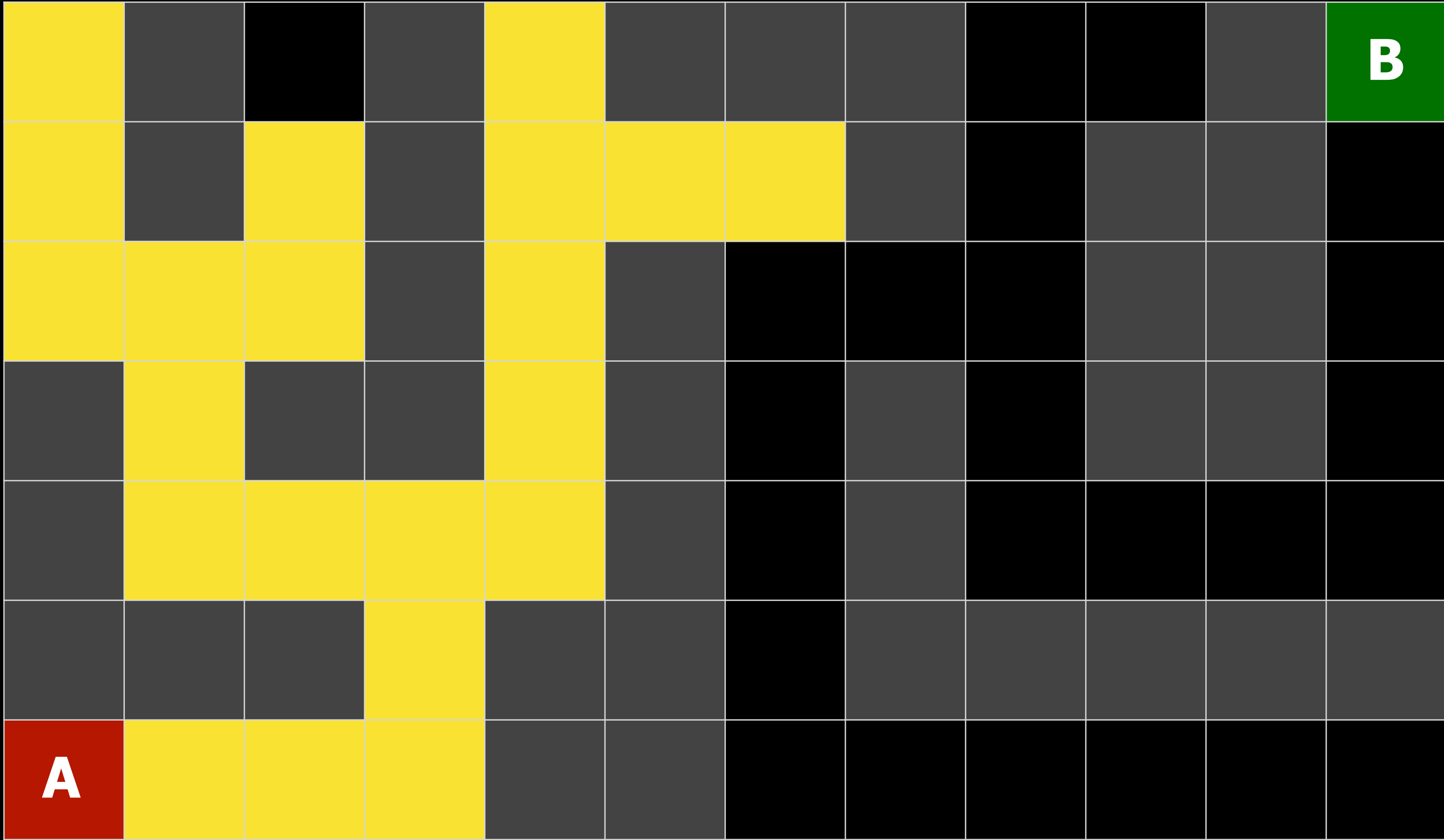
Breadth-First Search



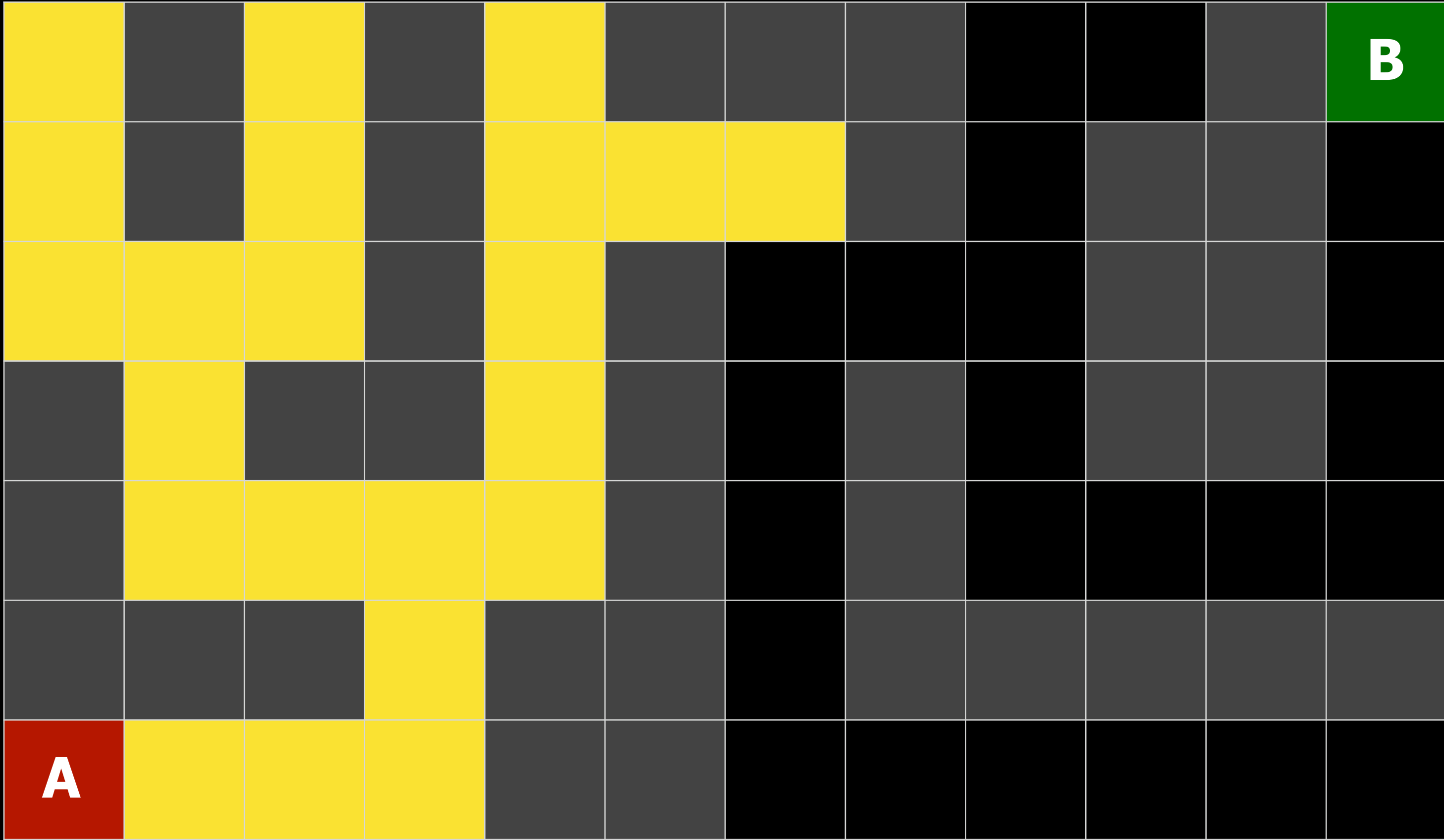
Breadth-First Search



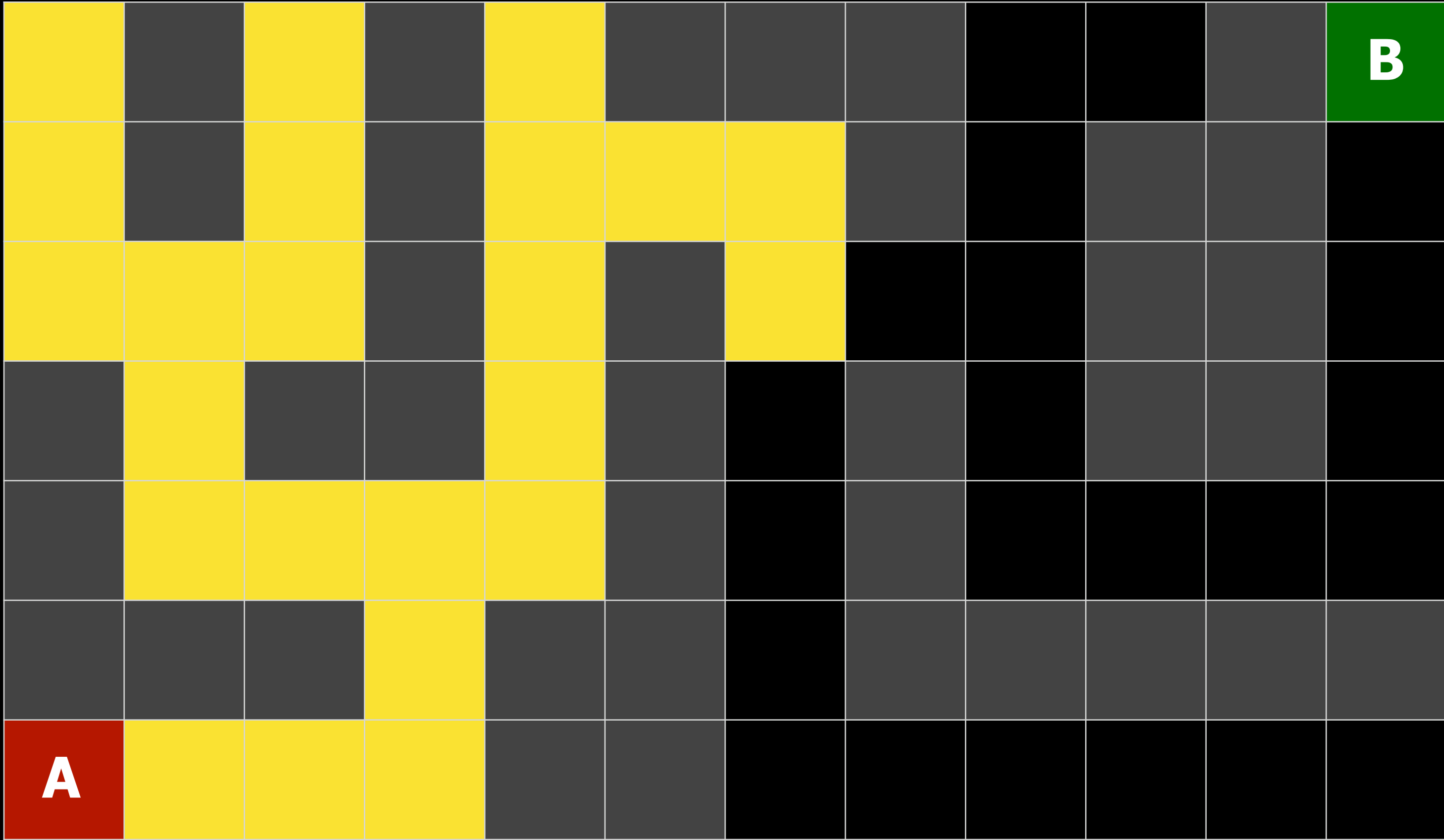
Breadth-First Search



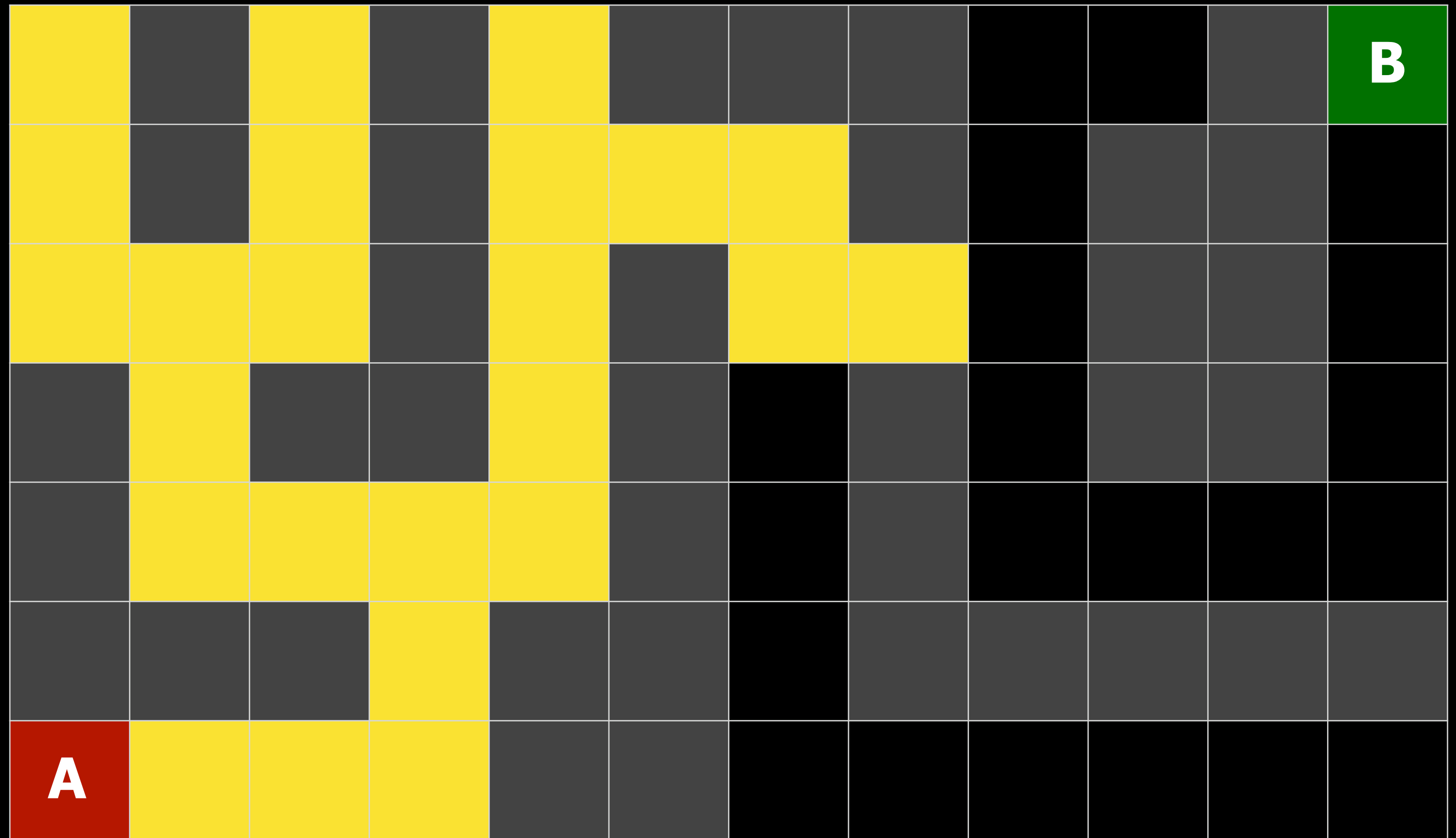
Breadth-First Search



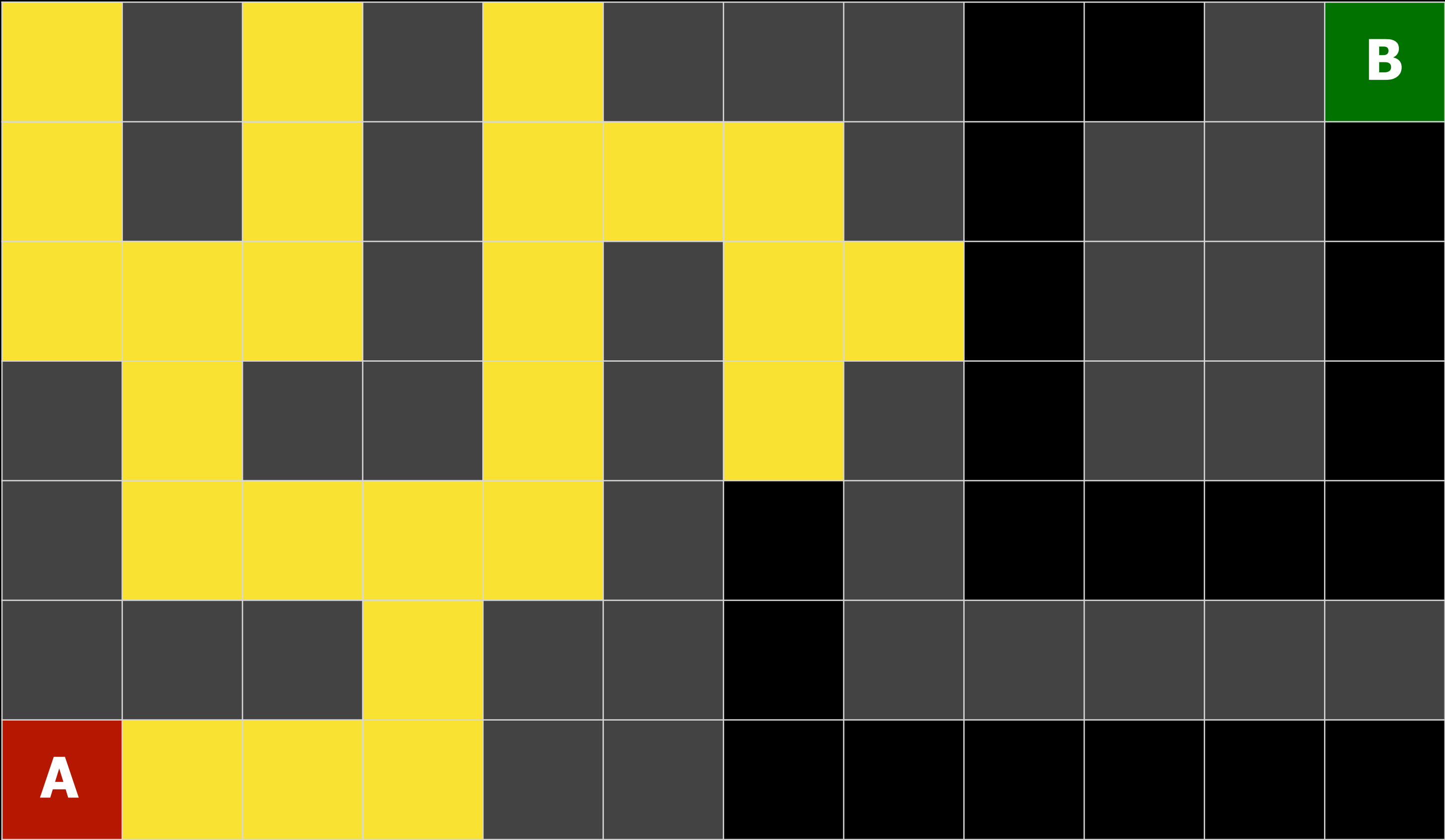
Breadth-First Search



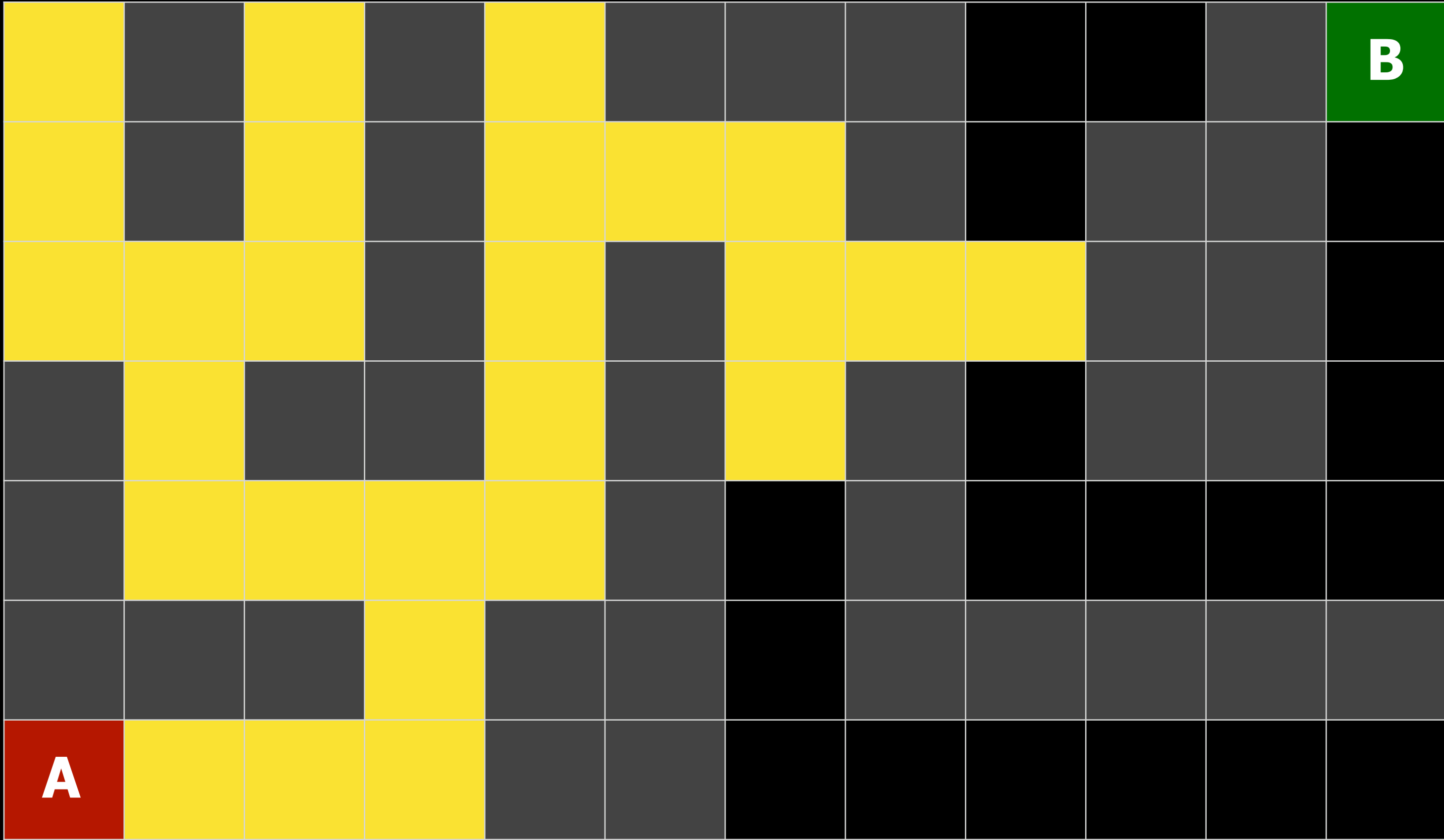
Breadth-First Search



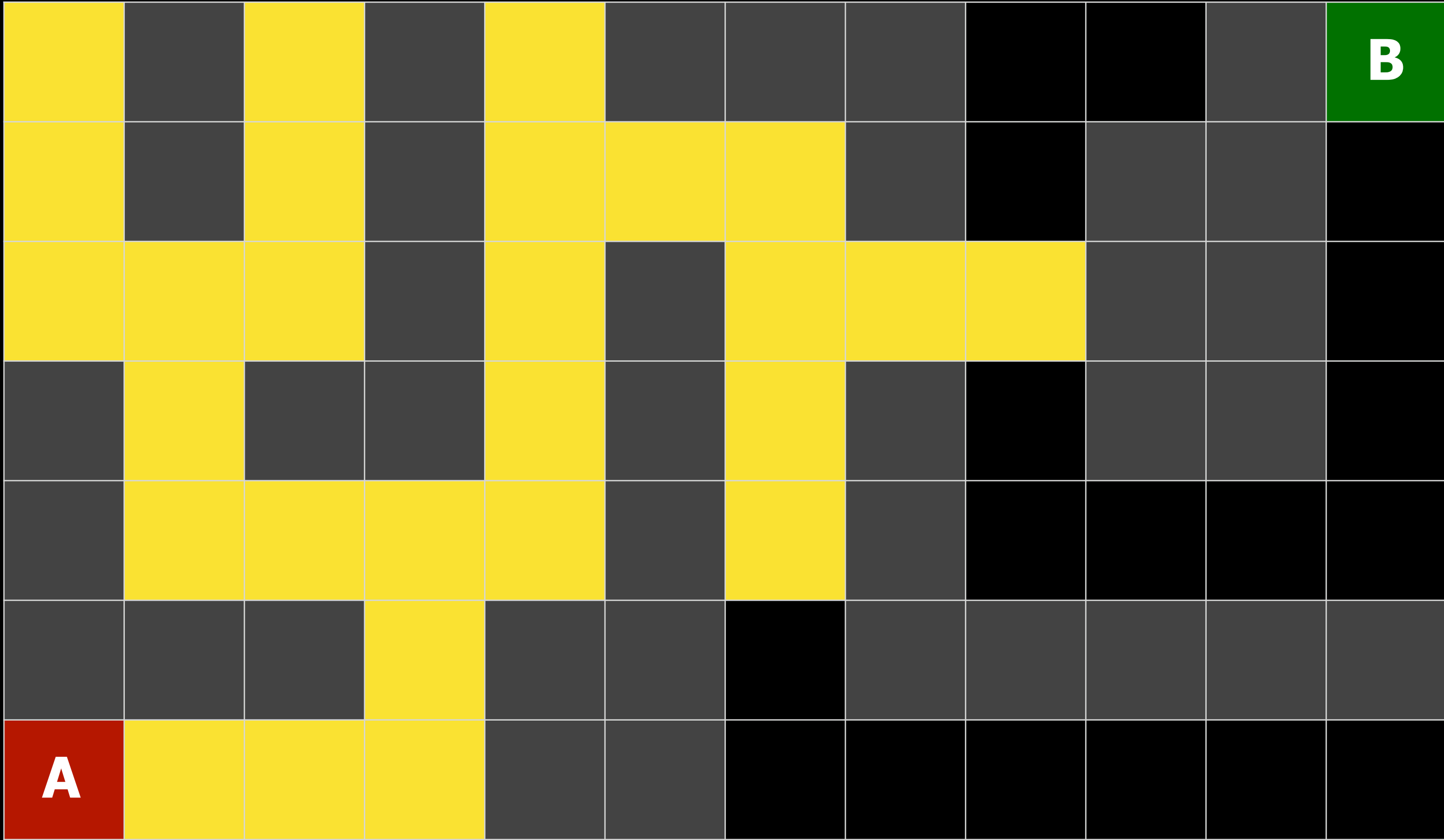
Breadth-First Search



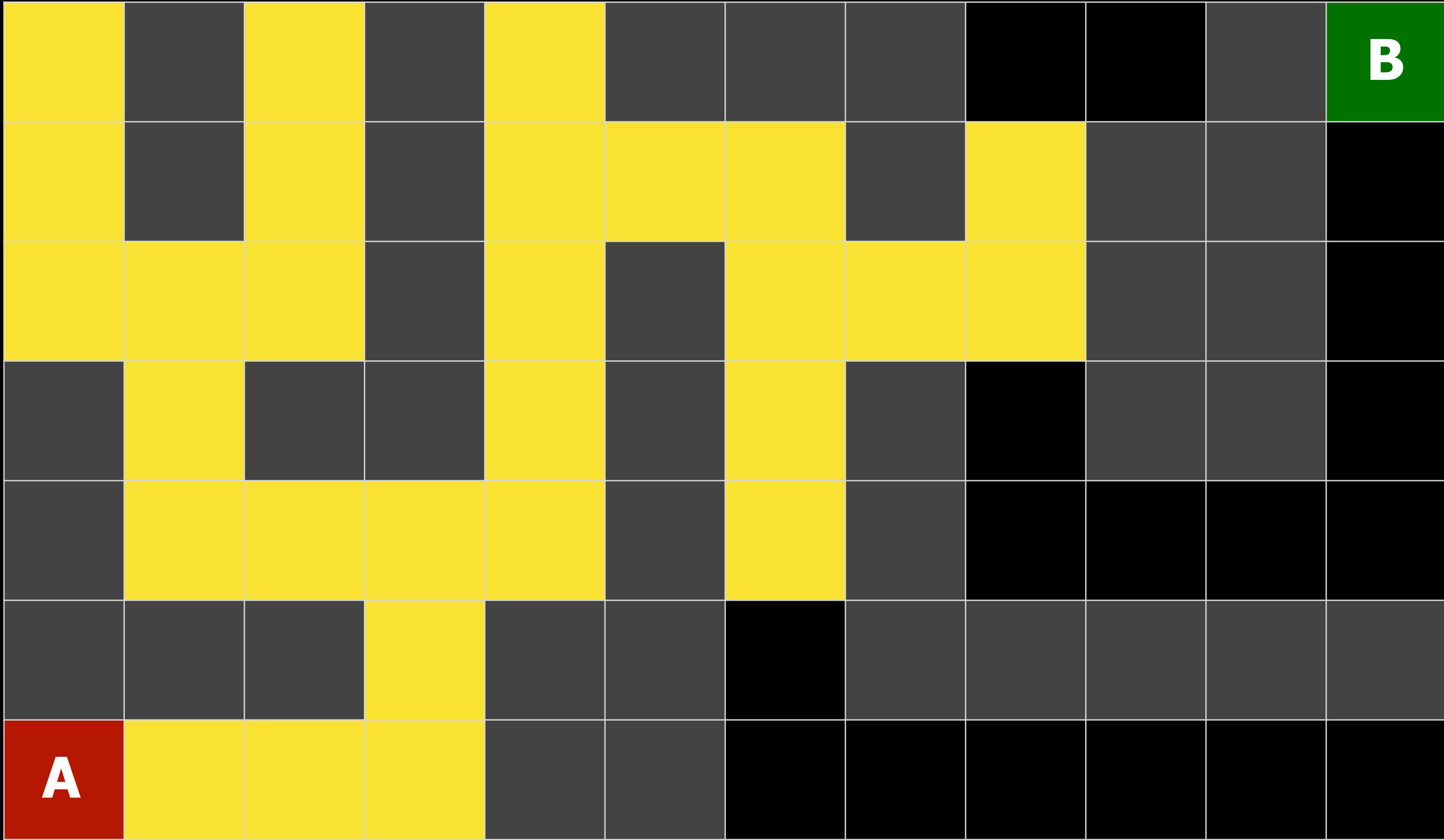
Breadth-First Search



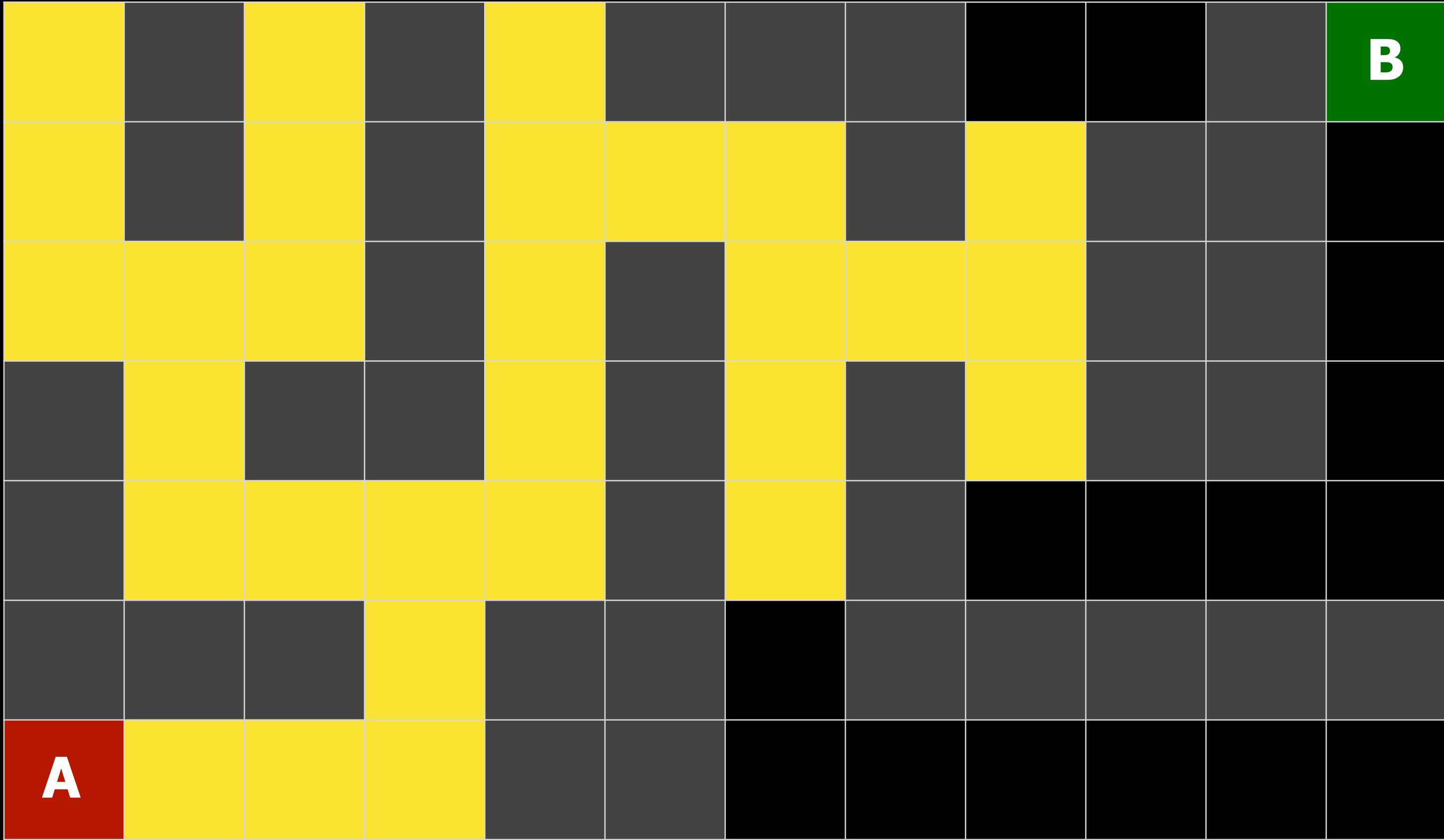
Breadth-First Search



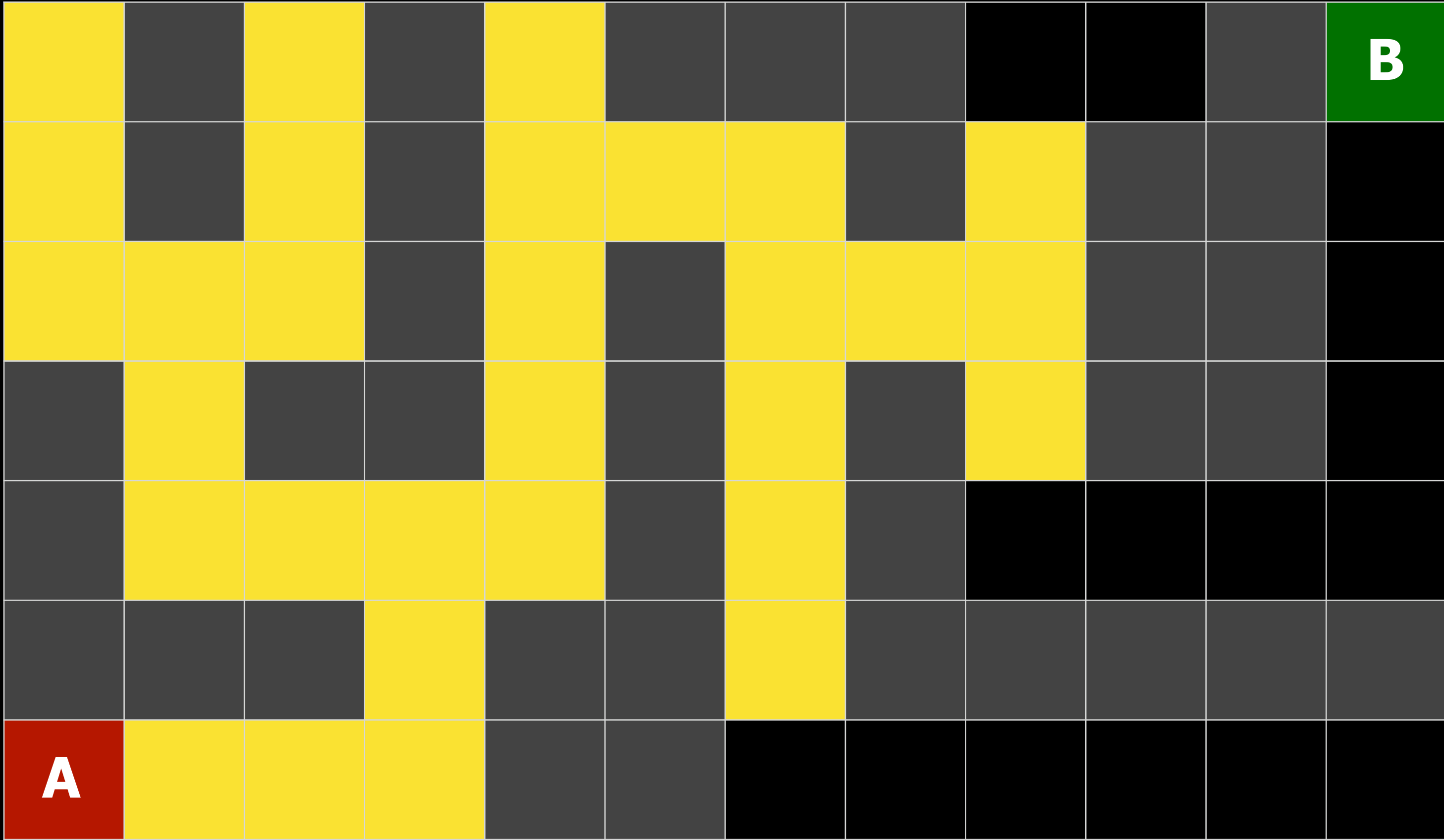
Breadth-First Search



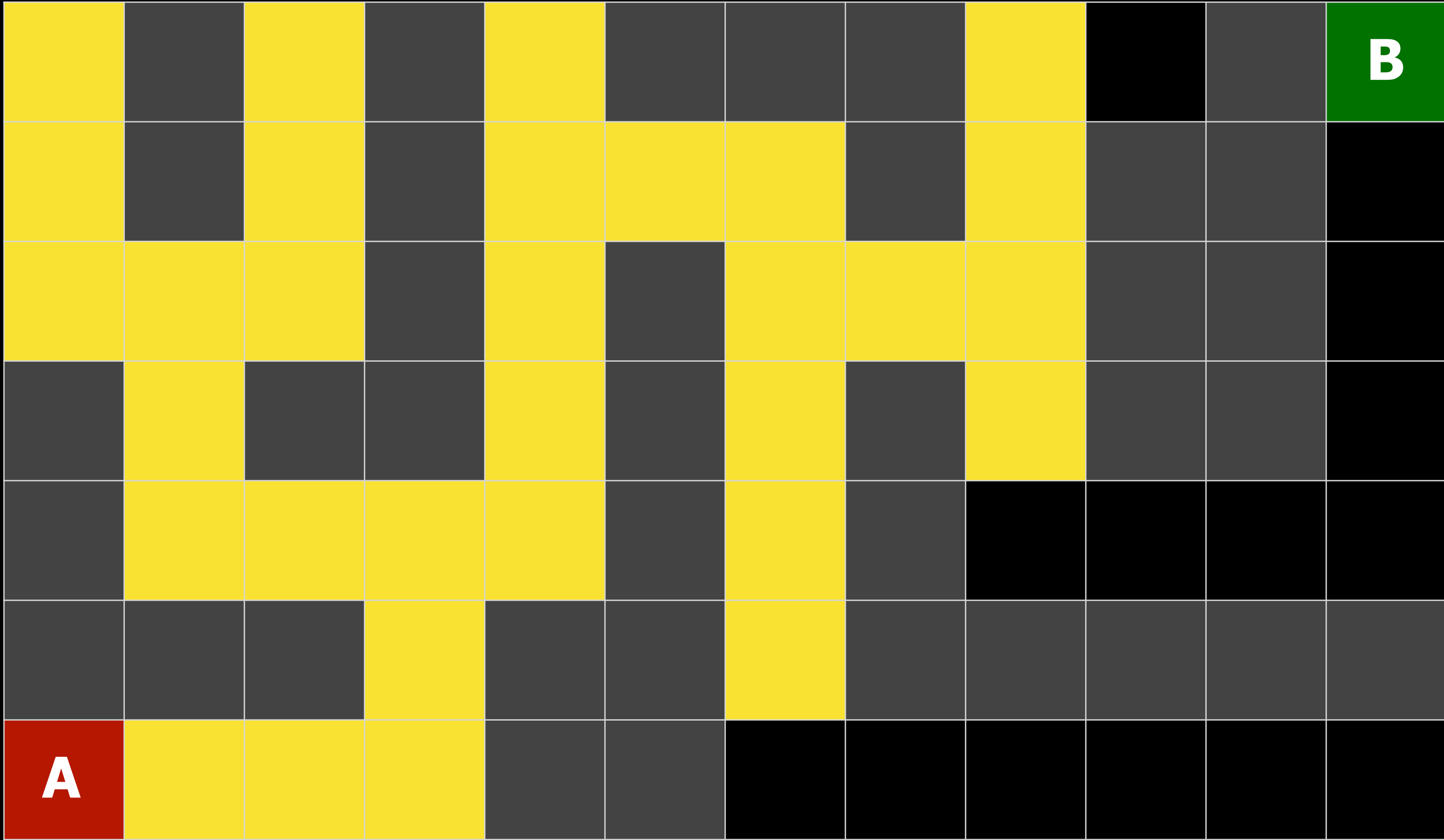
Breadth-First Search



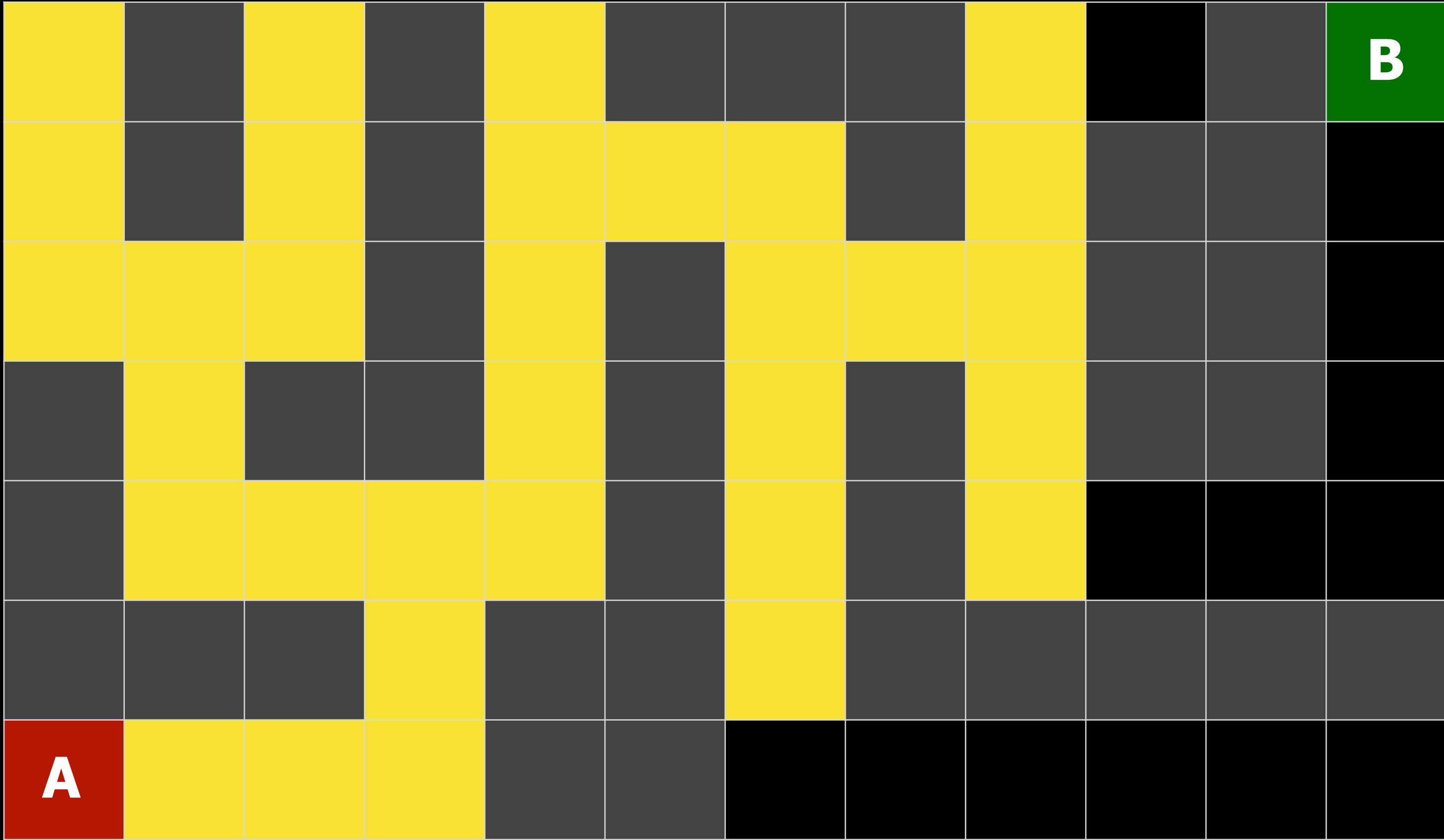
Breadth-First Search



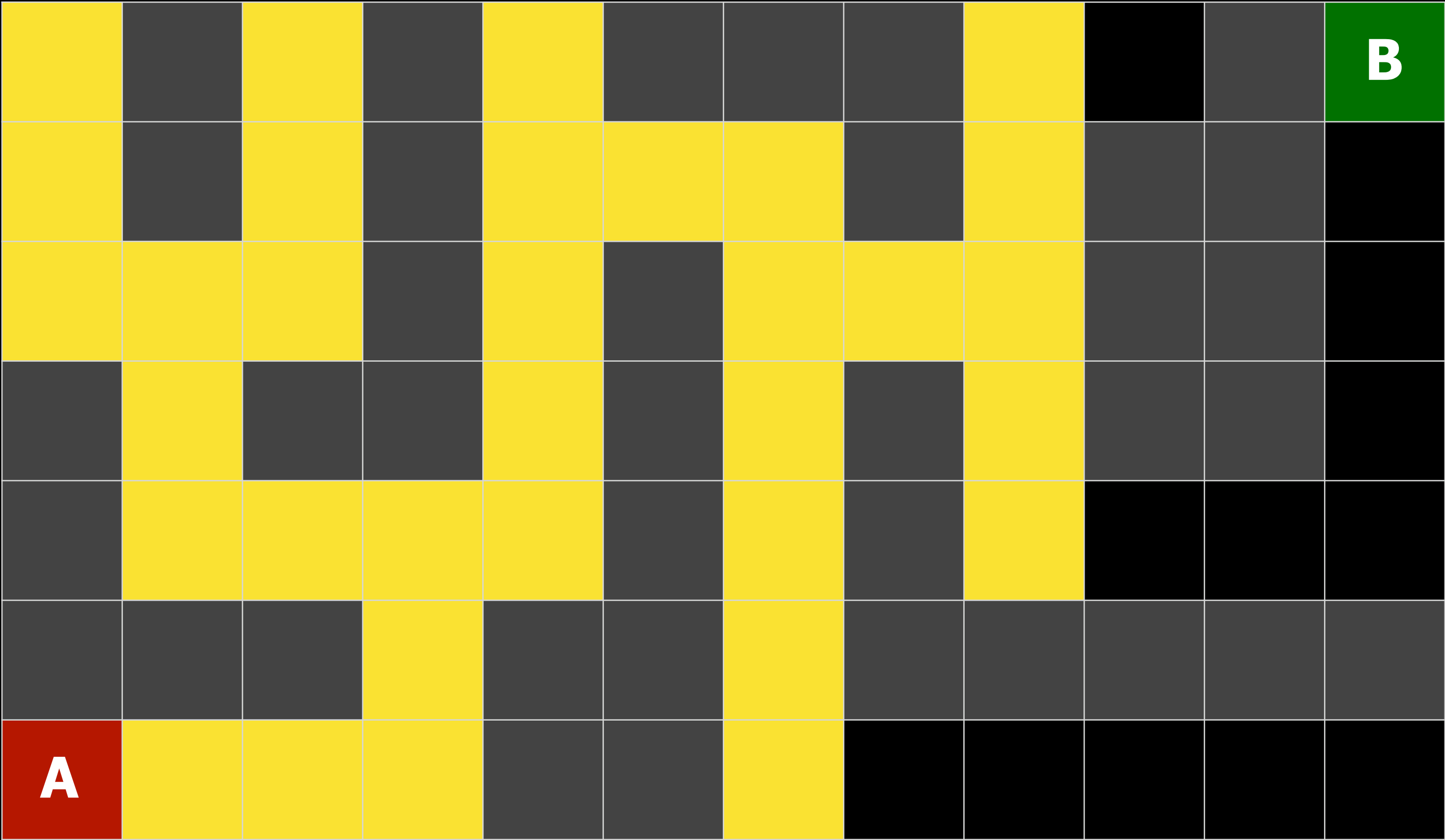
Breadth-First Search



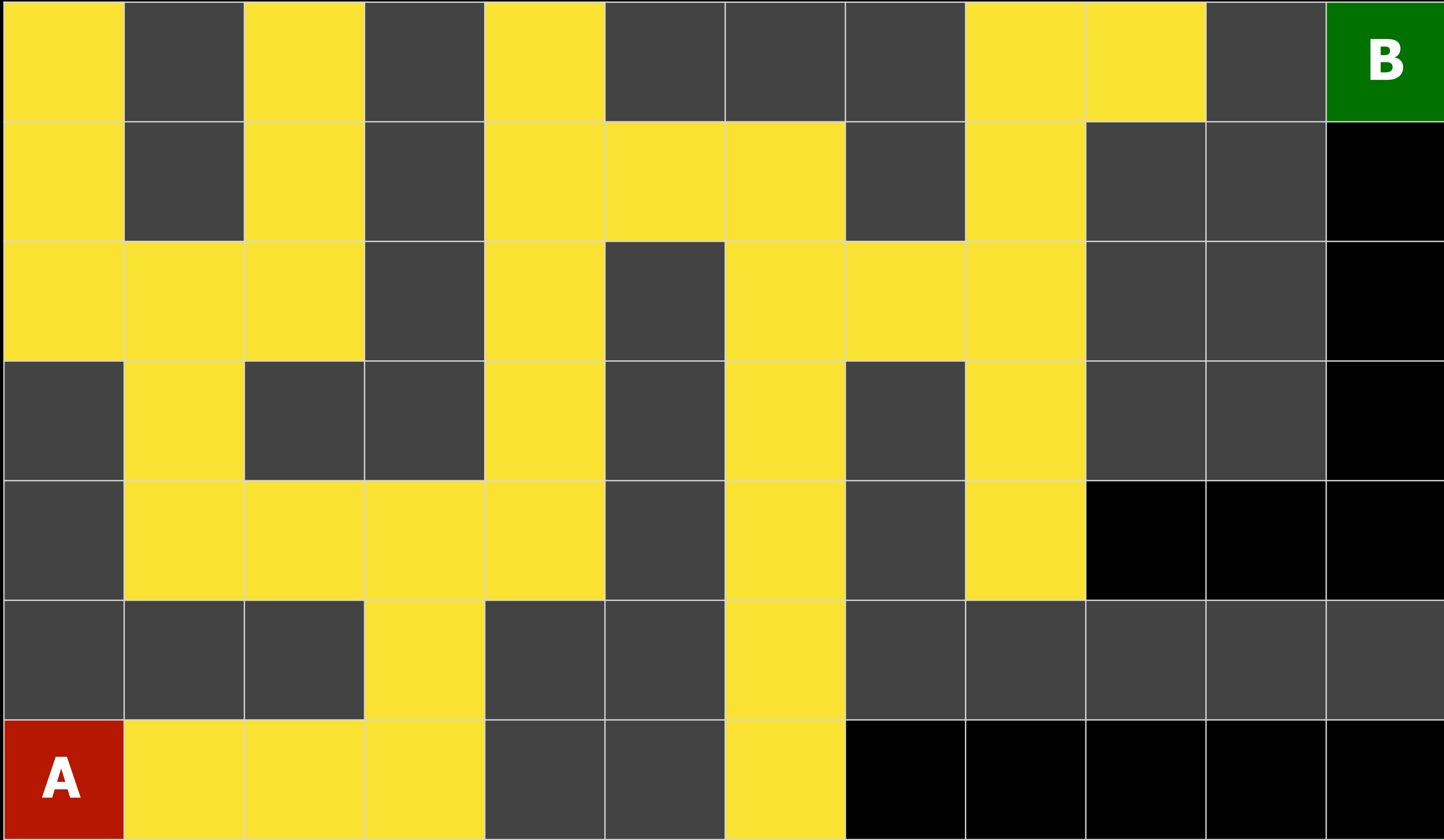
Breadth-First Search



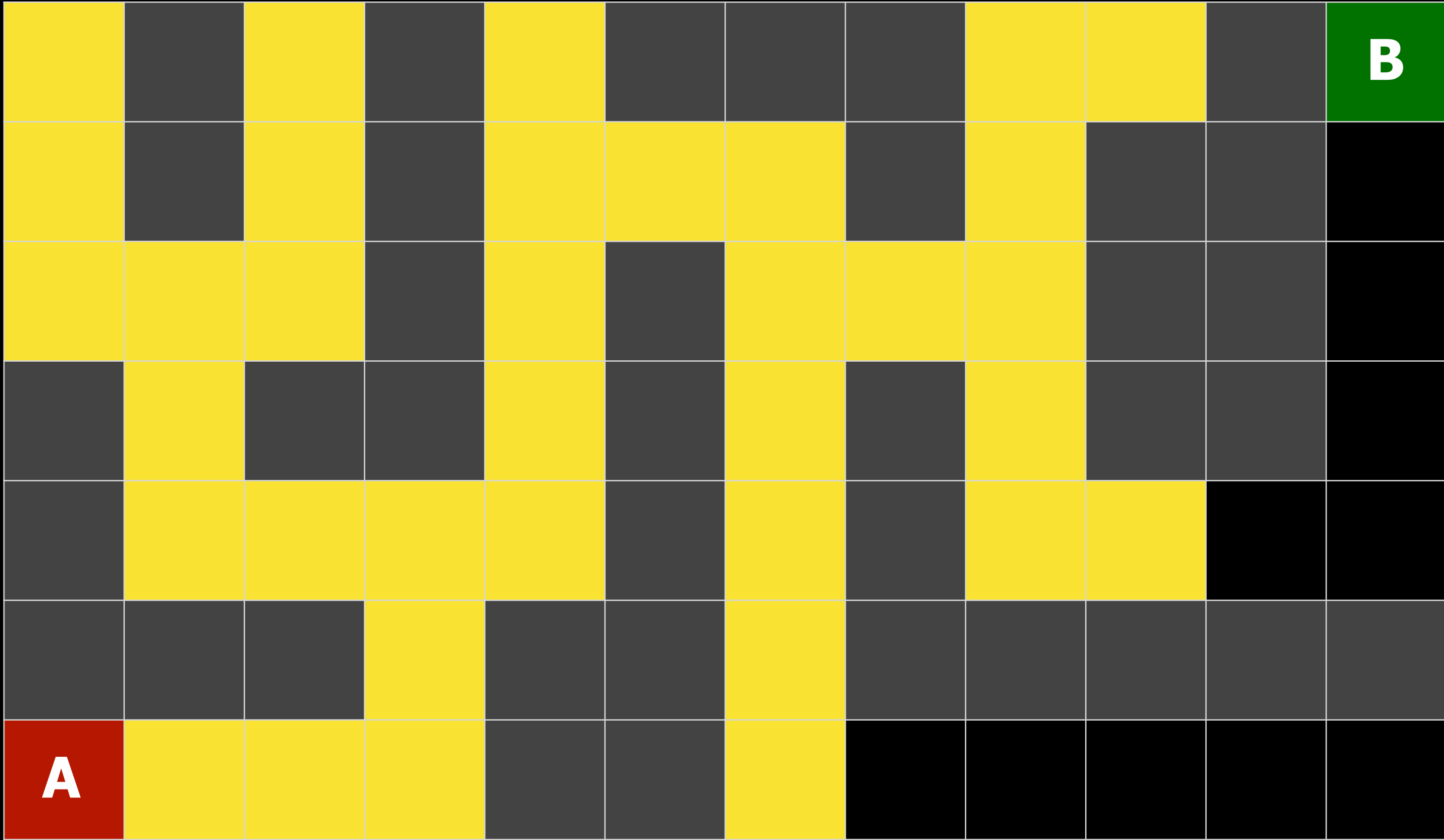
Breadth-First Search



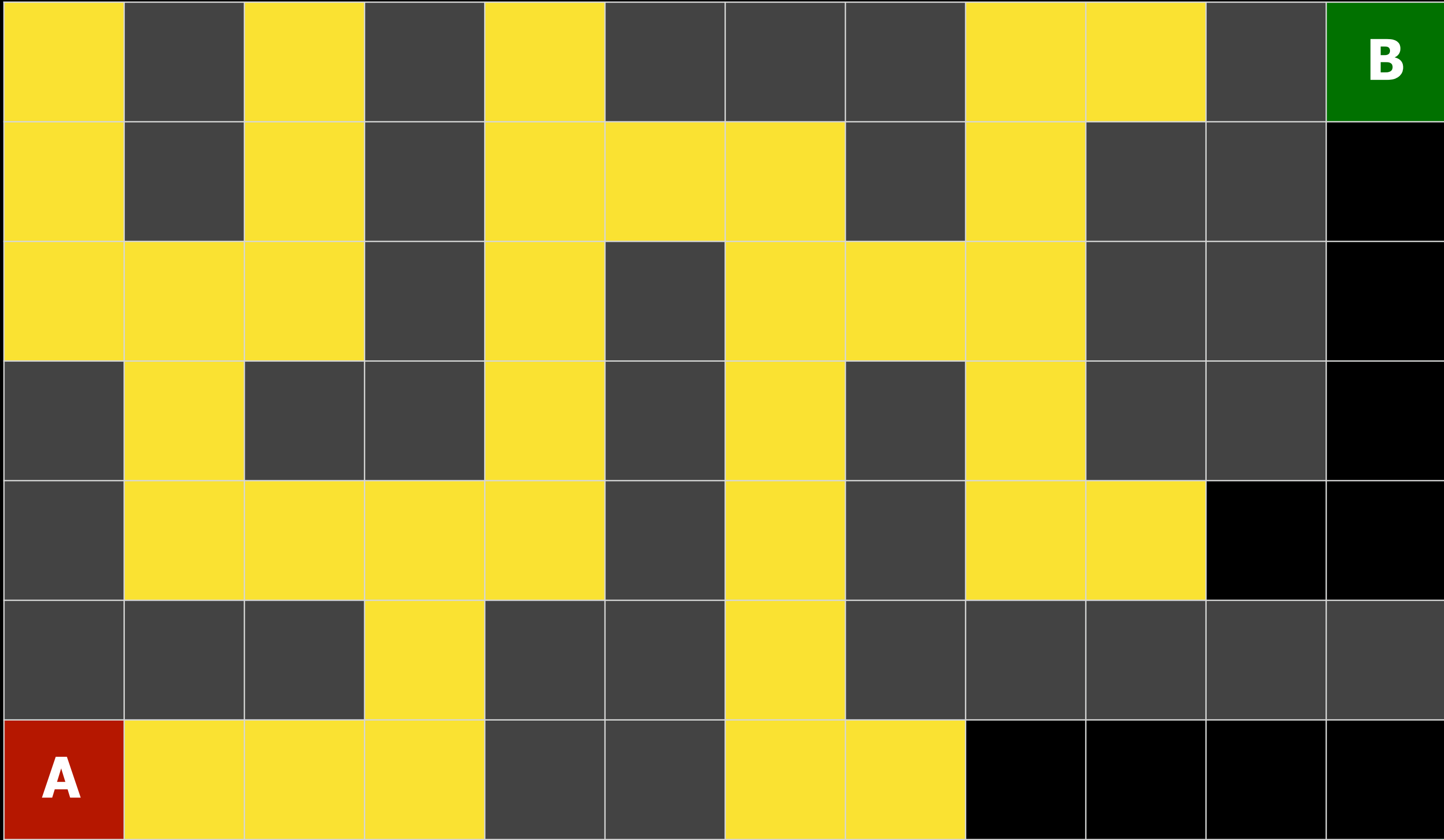
Breadth-First Search



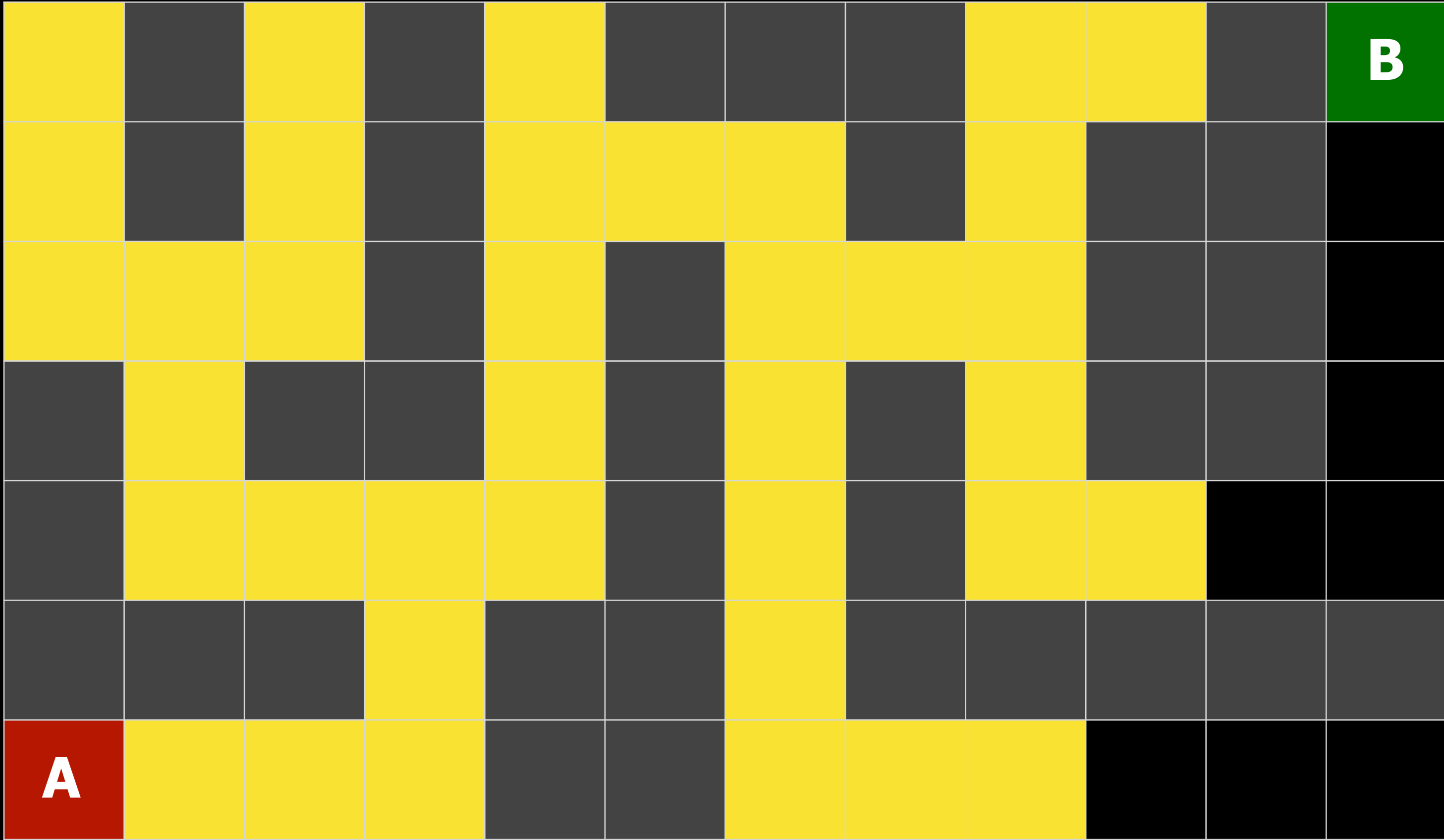
Breadth-First Search



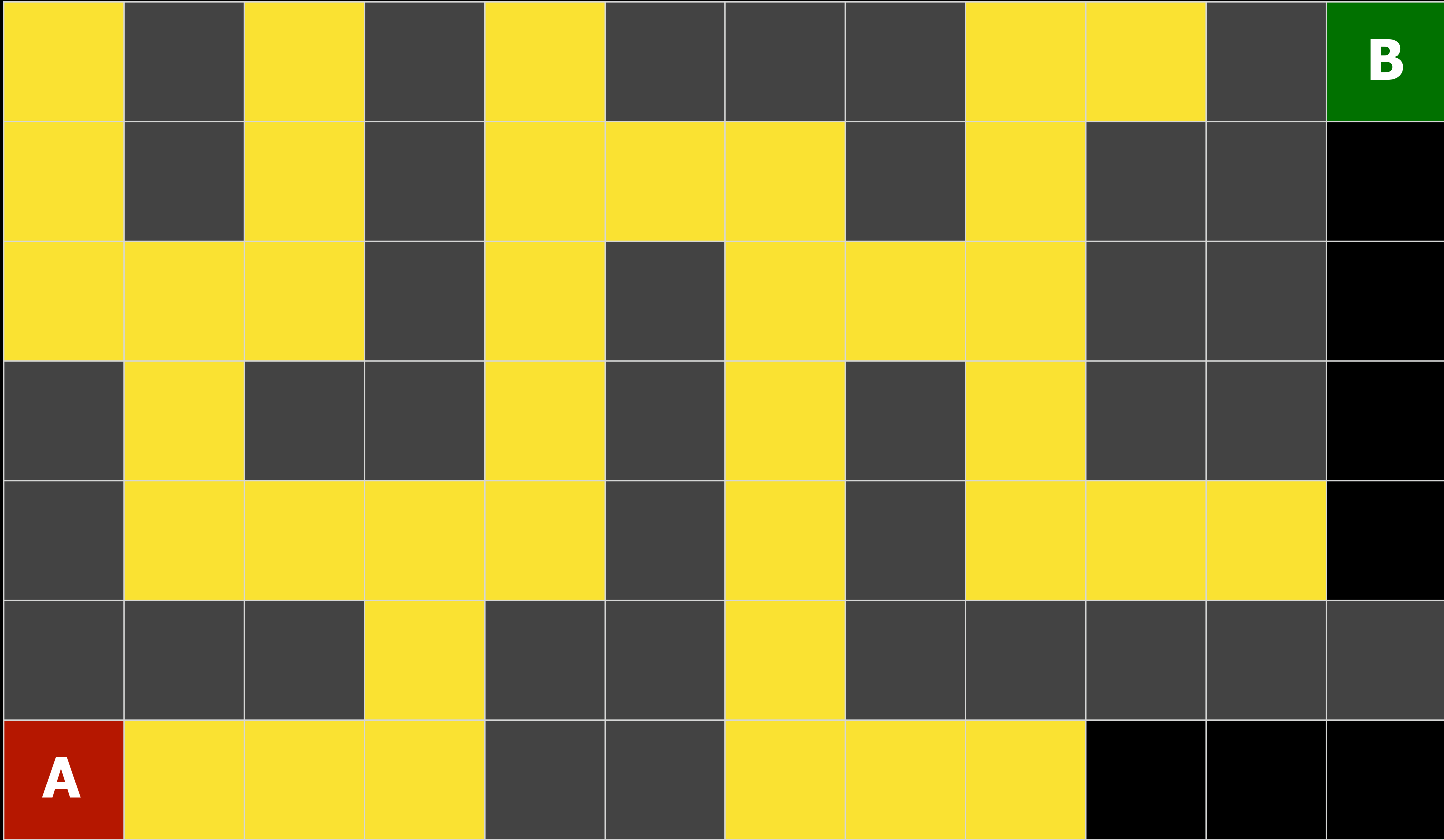
Breadth-First Search



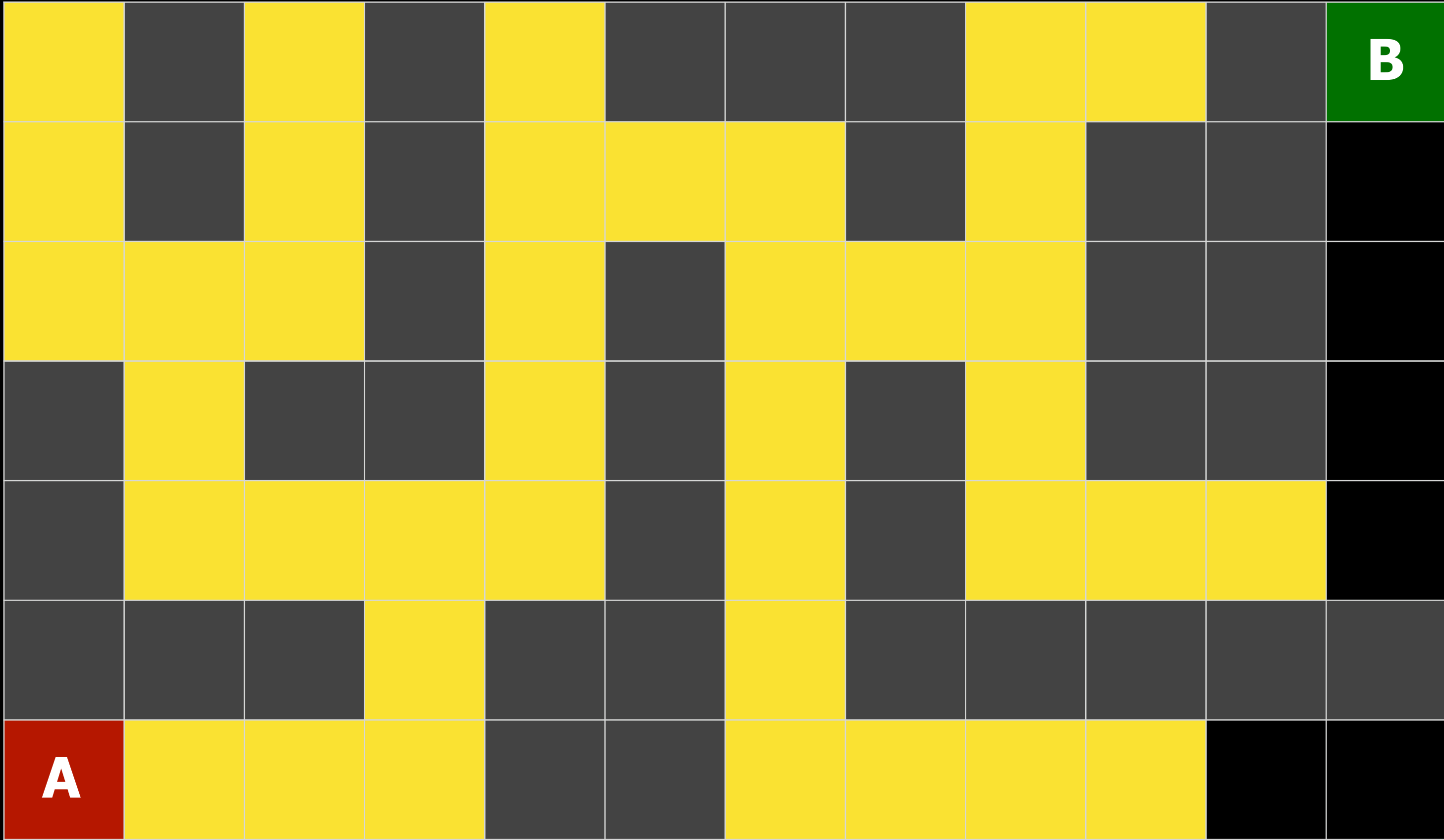
Breadth-First Search



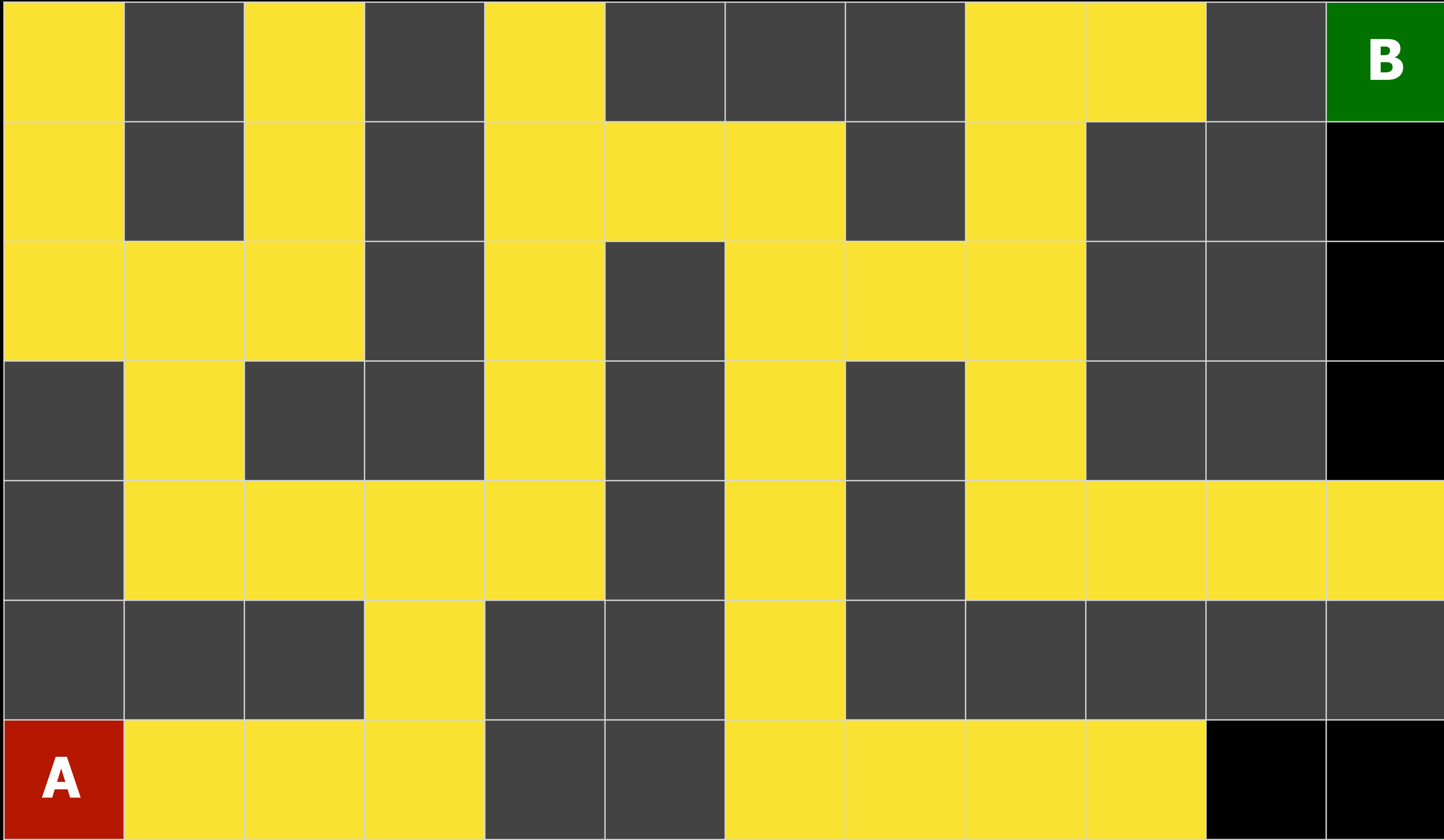
Breadth-First Search



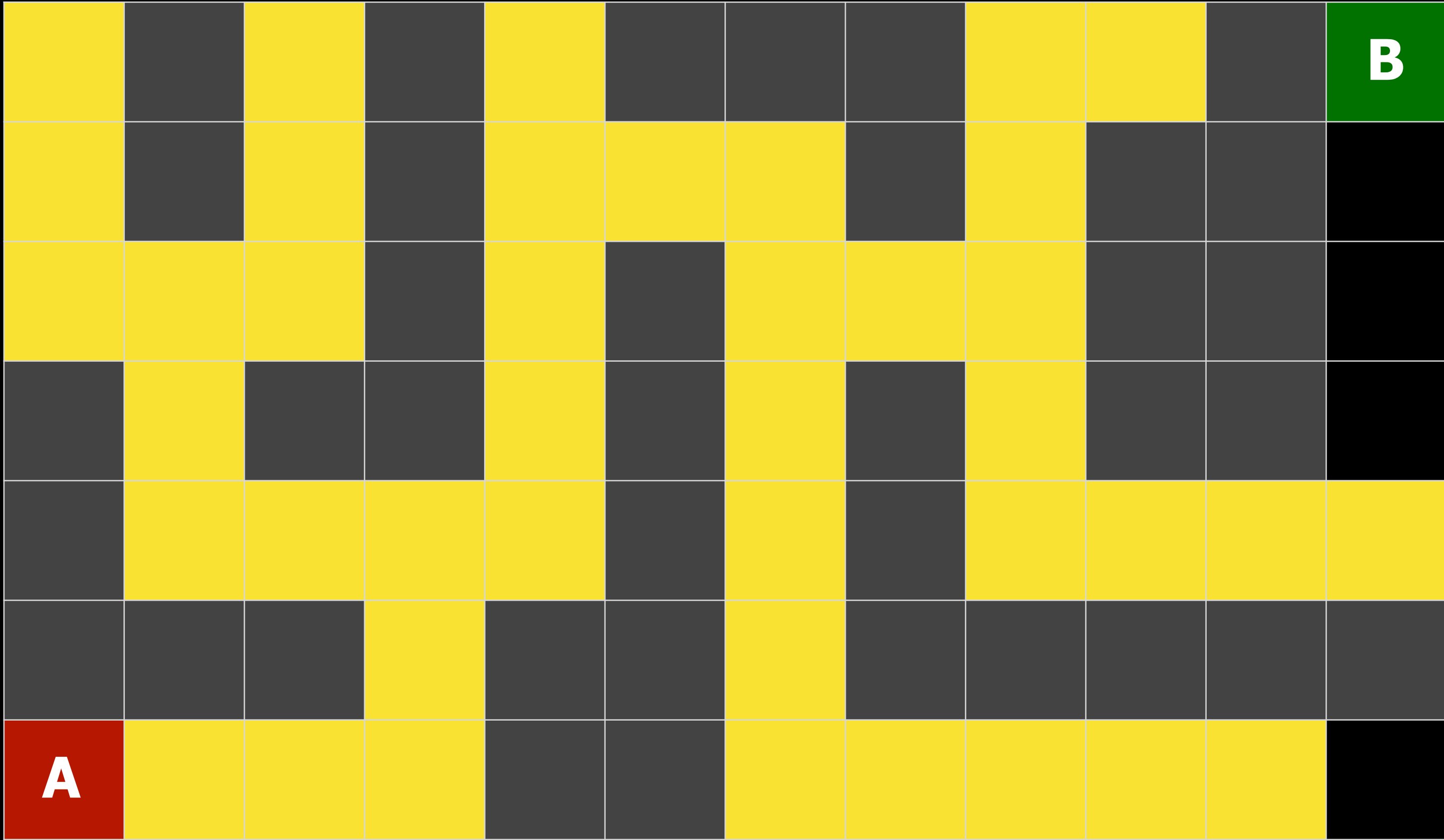
Breadth-First Search



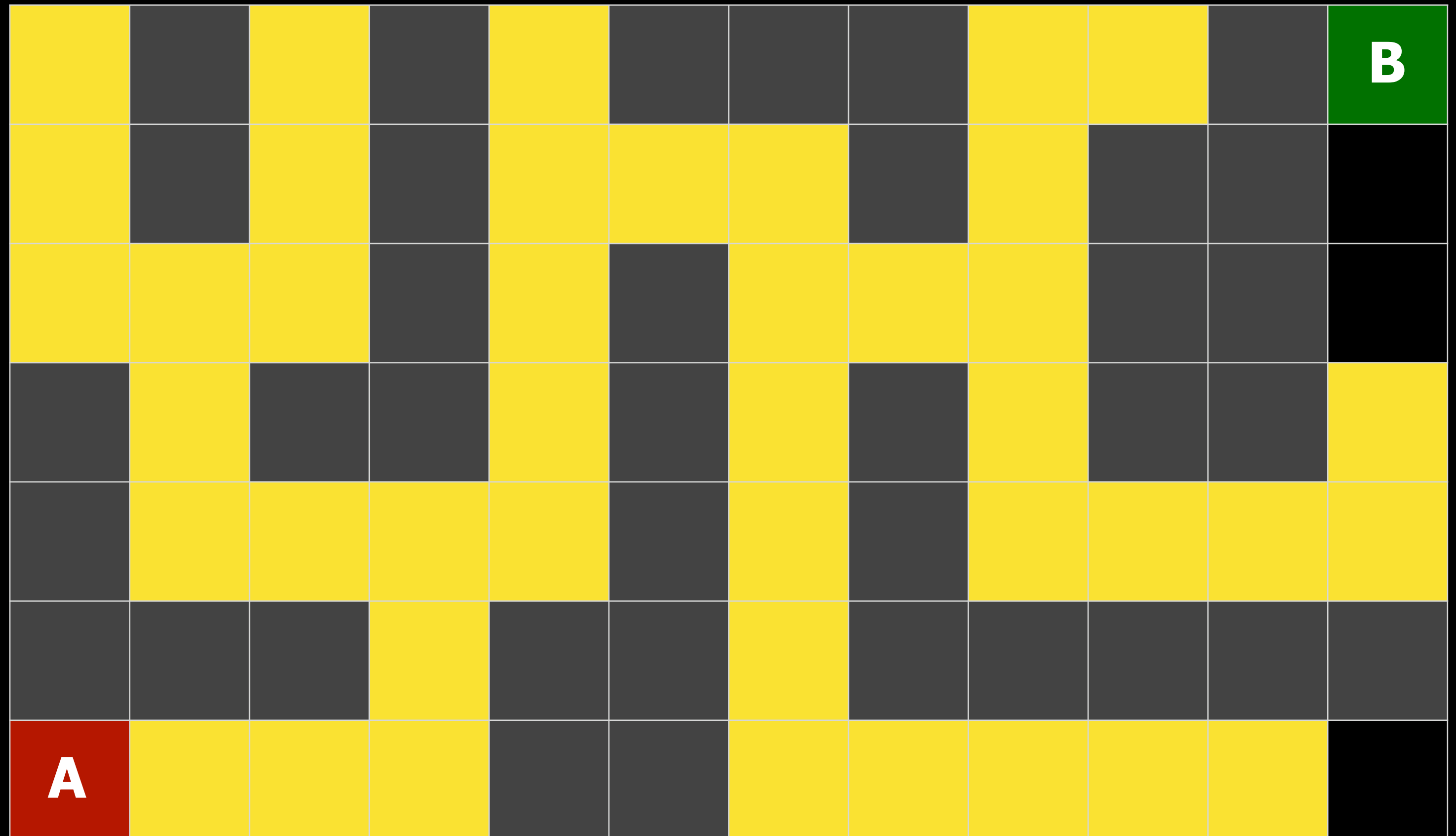
Breadth-First Search



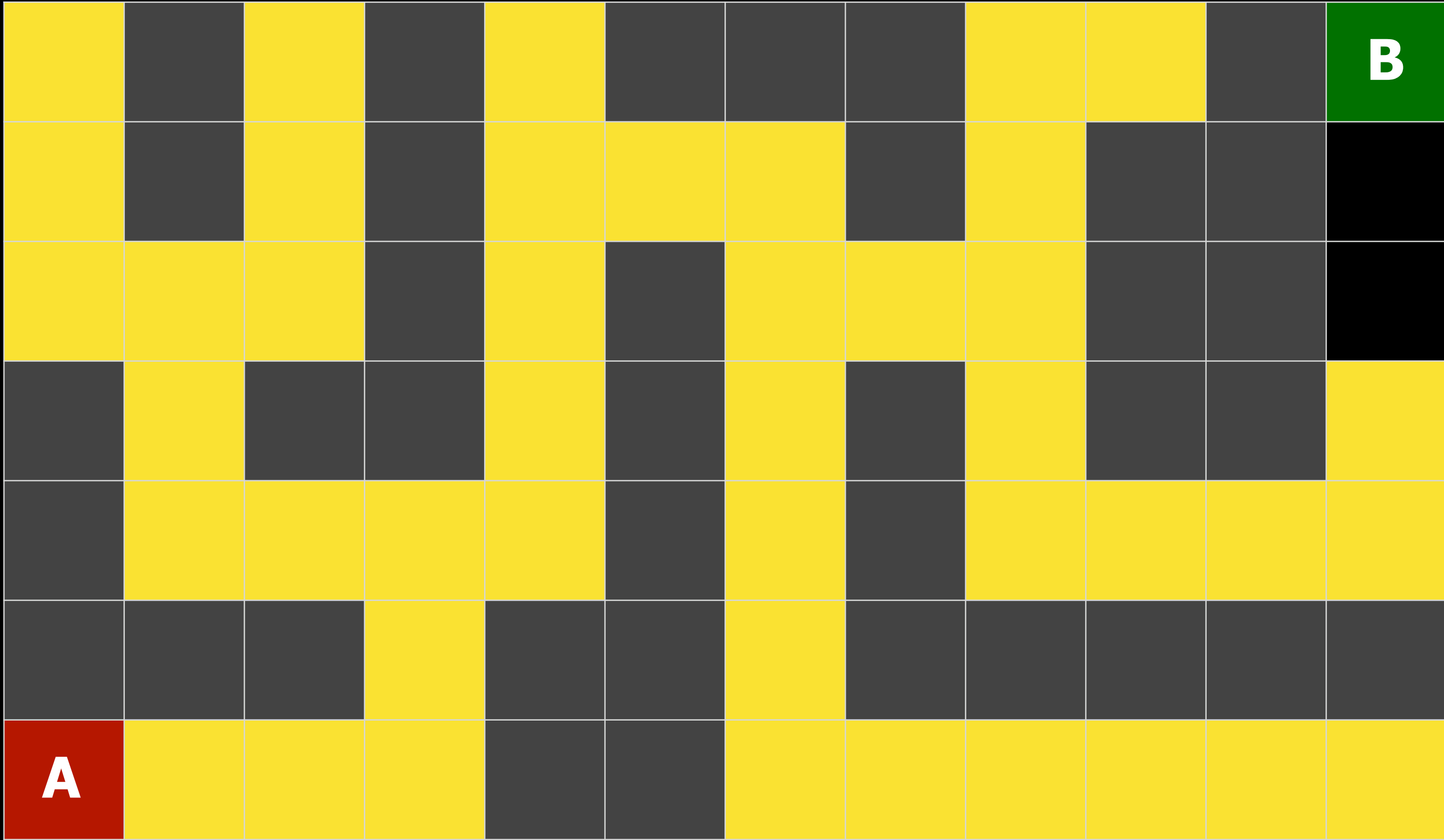
Breadth-First Search



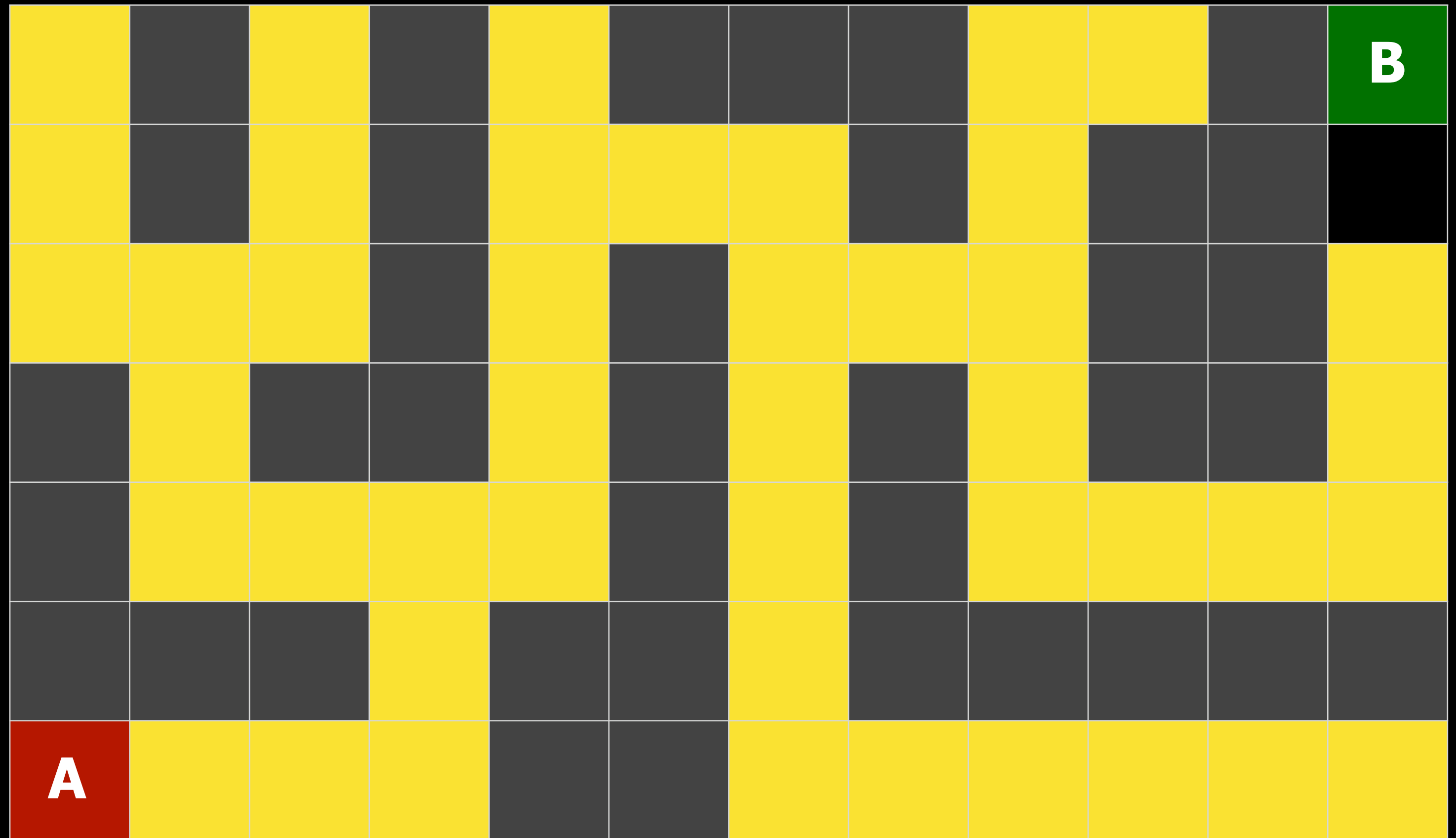
Breadth-First Search



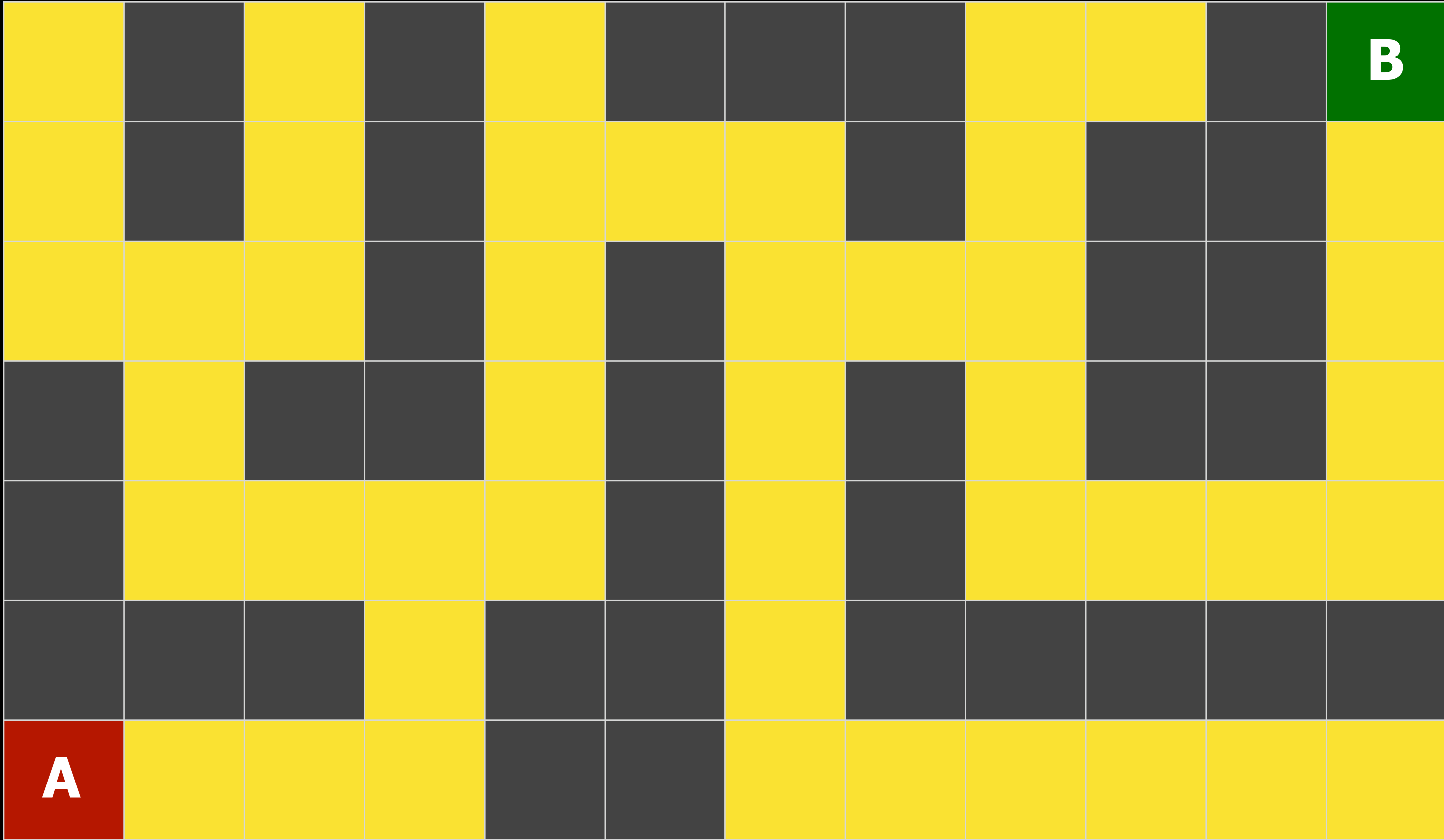
Breadth-First Search



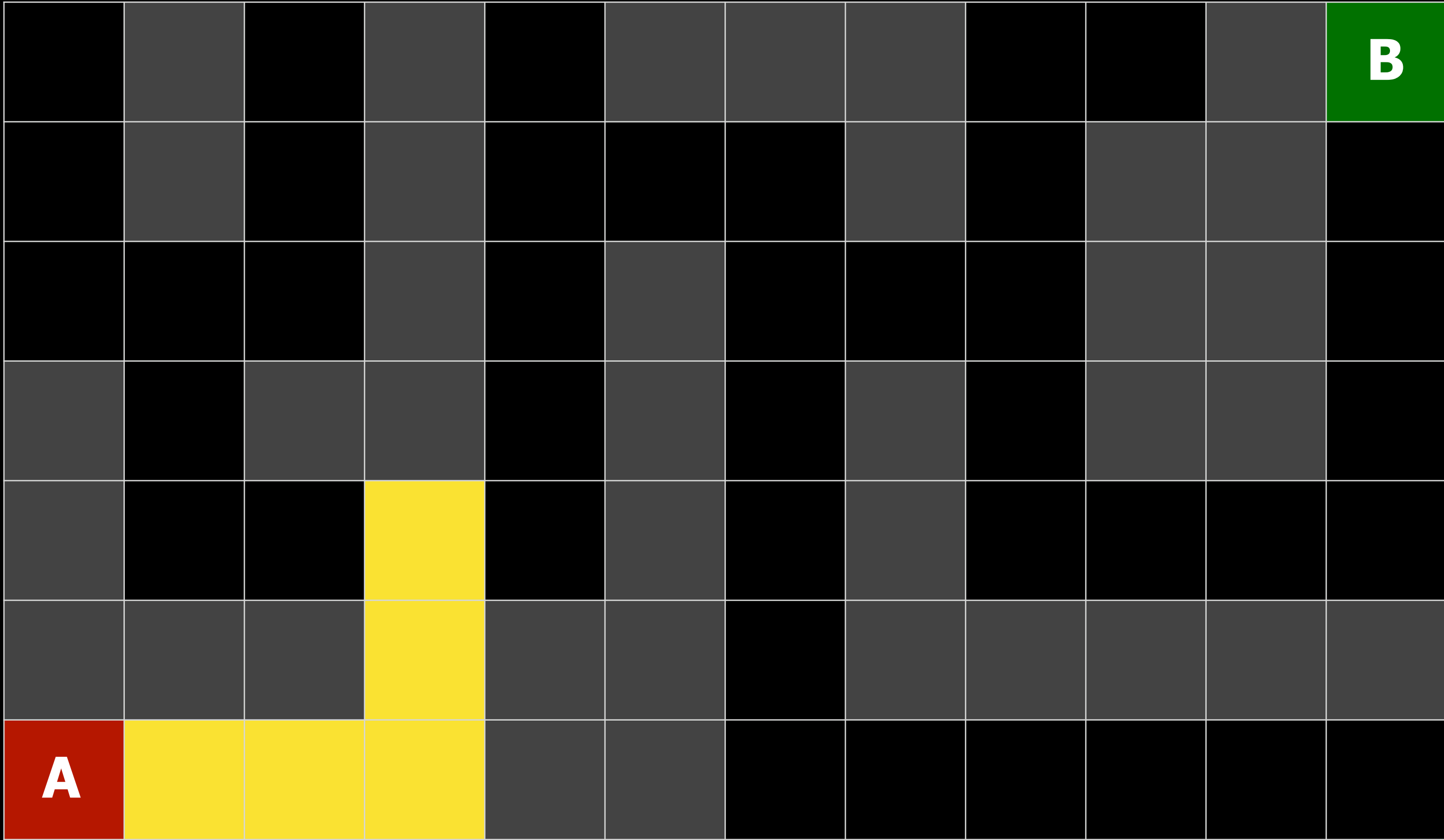
Breadth-First Search



Breadth-First Search



Breadth-First Search



Gracias!

lifeder
com

**Si quieres encontrar los secretos del
universo, piensa en términos de
energía, frecuencia y vibración**

-Nikola Tesla

AGENDA 29/08/2024

1. OBJETIVOS
2. RECAPITULACIÓN DE CLASE DEL 22/08/2024
3. VÍDEO REDES NEURONALES ESTRUCTURA
4. VÍDEO REDES NEURONALES CÓMO FUNCIONA
5. EJEMPLO PRÁCTICO DE REDES NEURONALES
6. USO DE GIT Y RESUMENES DE CLASE
7. INTRODUCCIÓN A CÓDIGO EN PYTHON PARA BÚSQUEDA
8. PROPOSICIONES

OBJETIVOS

COMPRENDER EL FUNCIONAMIENTO INTERNO DE UNA RED NEURONAL EN CUANTO A SU ESTRUCTURA
PRACTICAR EL USO DE GITHUB Y LOS COMANDOS DE GIT PARA REALIZAR RESUMENES
ENTENDER EL FUNCIONAMIENTO DE UN ALGORITMO DE BUSQUEDA IMPLEMENTADO EN PYTHON

REDES NEURONALES

ESTRUCTURA DE RED NEURONAL

<https://www.youtube.com/watch?v=aircAruvnKk>

REDES NEURONALES

COMO FUNCIONAN

https://www.youtube.com/watch?v=IHZwWFHWa-w&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=2

EJEMPLO PRÁCTICO DE USO DE RED NEURONAL

INICIO DE RED NEURONAL EJERCICIOS

https://www.youtube.com/watch?v=iX_on3VxZzk&t=47s

<https://www.youtube.com/watch?v=UNFFLJPW7KQ>

USO DE GIT Y GITHUB

GUIAS EN INTERNET Y PDF

<https://githubtraining.github.io/training-manual/book.pdf>

https://training.github.com/downloads/es_ES/github-git-cheat-sheet.pdf

<https://www.youtube.com/watch?v=RGQj5yH7evk>

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

<https://www.atlassian.com/git/tutorials/install-git>

INTRODUCCION AL ALGORITMO DE BUSQUEDA EN PYTHON

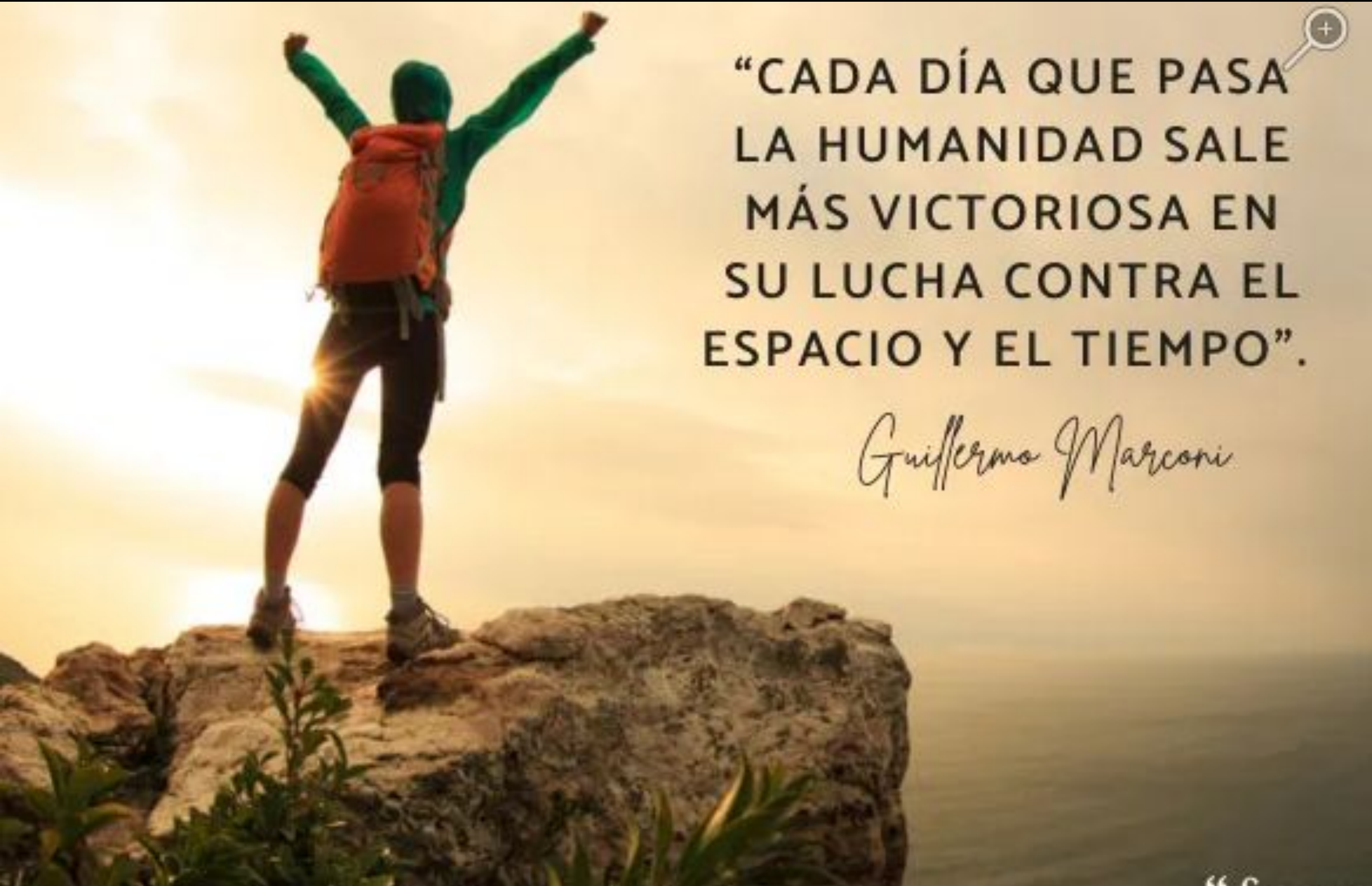
Revisión de documento

PROPOSICIONES

Hasta el momento hay propuestas para
la metodología de clase?
Dificultades?

afecreer@gmail.com

GRACIAS



“CADA DÍA QUE PASA
LA HUMANIDAD SALE
MÁS VICTORIOSA EN
SU LUCHA CONTRA EL
ESPACIO Y EL TIEMPO”.

Guillermo Marconi

AGENDA 12/08/2024

1. OBJETIVOS
2. RECAPITULACIÓN DE CLASE DEL 29/08/2024
3. VÍDEO REDES NEURONALES ESTRUCTURA
4. VÍDEO REDES NEURONALES CÓMO FUNCIONA
5. EJEMPLO PRÁCTICO DE REDES NEURONALES
6. USO DE GIT Y RESUMENES DE CLASE
7. INTRODUCCIÓN A CÓDIGO EN PYTHON PARA BÚSQUEDA
8. PROPOSICIONES

Búsqueda No informada

Búsqueda Informada

Es una estrategia de búsqueda que usa conocimiento específico del problema para encontrar una solución mas eficiente

Algoritmo codicioso de busqueda del mejor primero

Este algoritmo expande el nodo,
que está más cercano a la meta lo
cual se estima con una función
heurística $h(n)$

Algoritmo codicioso función heurística $h(n)$

En el problema del 8-Puzzle (donde se debe mover fichas dentro de una cuadrícula hasta alcanzar una configuración objetivo), se puede usar una heurística como:

1. **Distancia Manhattan:** Mide la suma de las distancias de cada ficha desde su posición actual hasta su posición objetivo, sumando solo los movimientos horizontales y verticales.

$$h(n) = \sum_i |x_i - x_{i,goal}| + |y_i - y_{i,goal}|$$

Donde (x_i, y_i) es la posición actual de la ficha i y $(x_{\{i, goal\}}, y_{\{i, goal\}})$ es la posición objetivo de la ficha i .

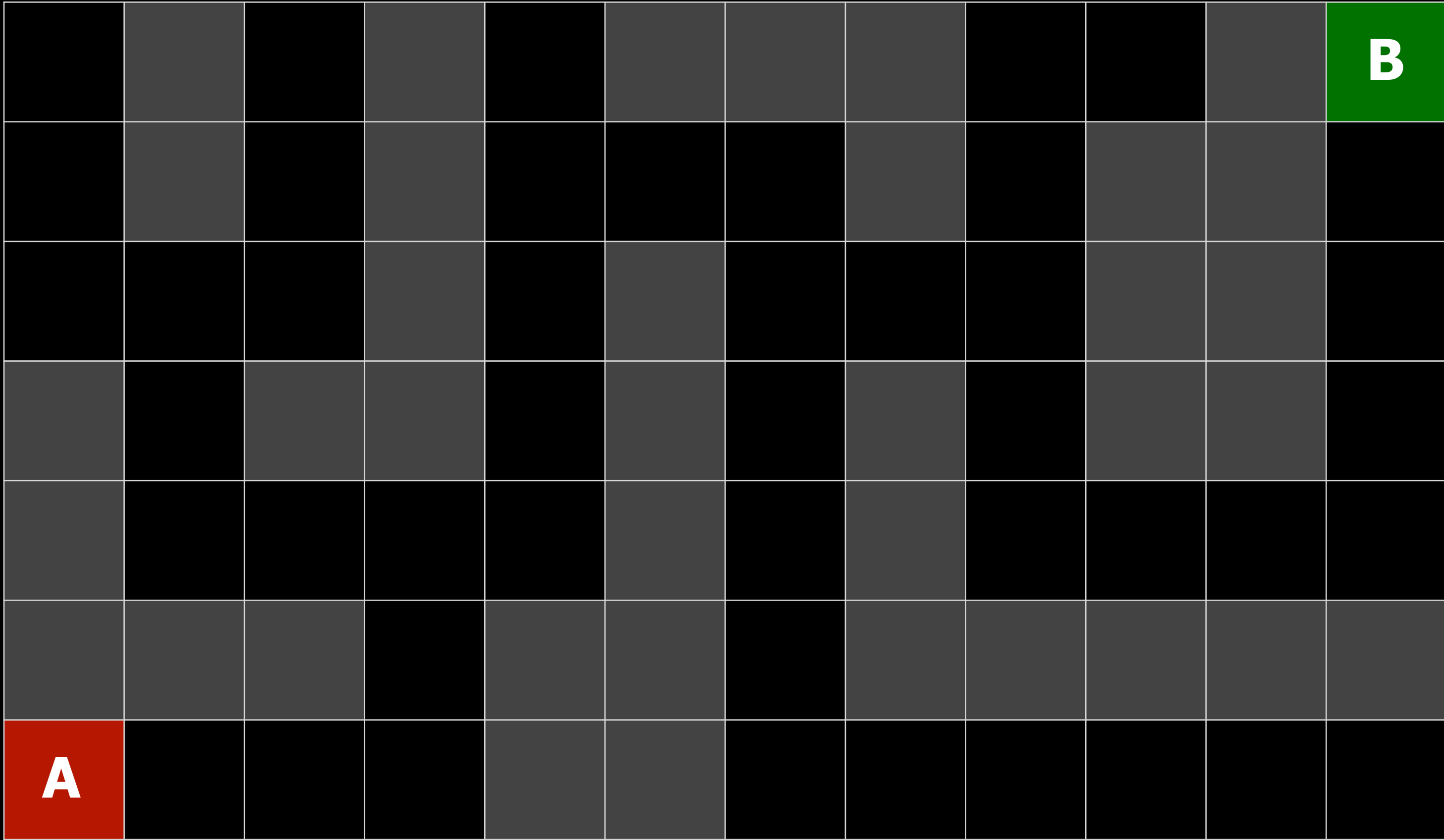
Esta función proporciona una estimación del número mínimo de movimientos necesarios para resolver el rompecabezas.

Algoritmo codicioso función heurística $h(n)$

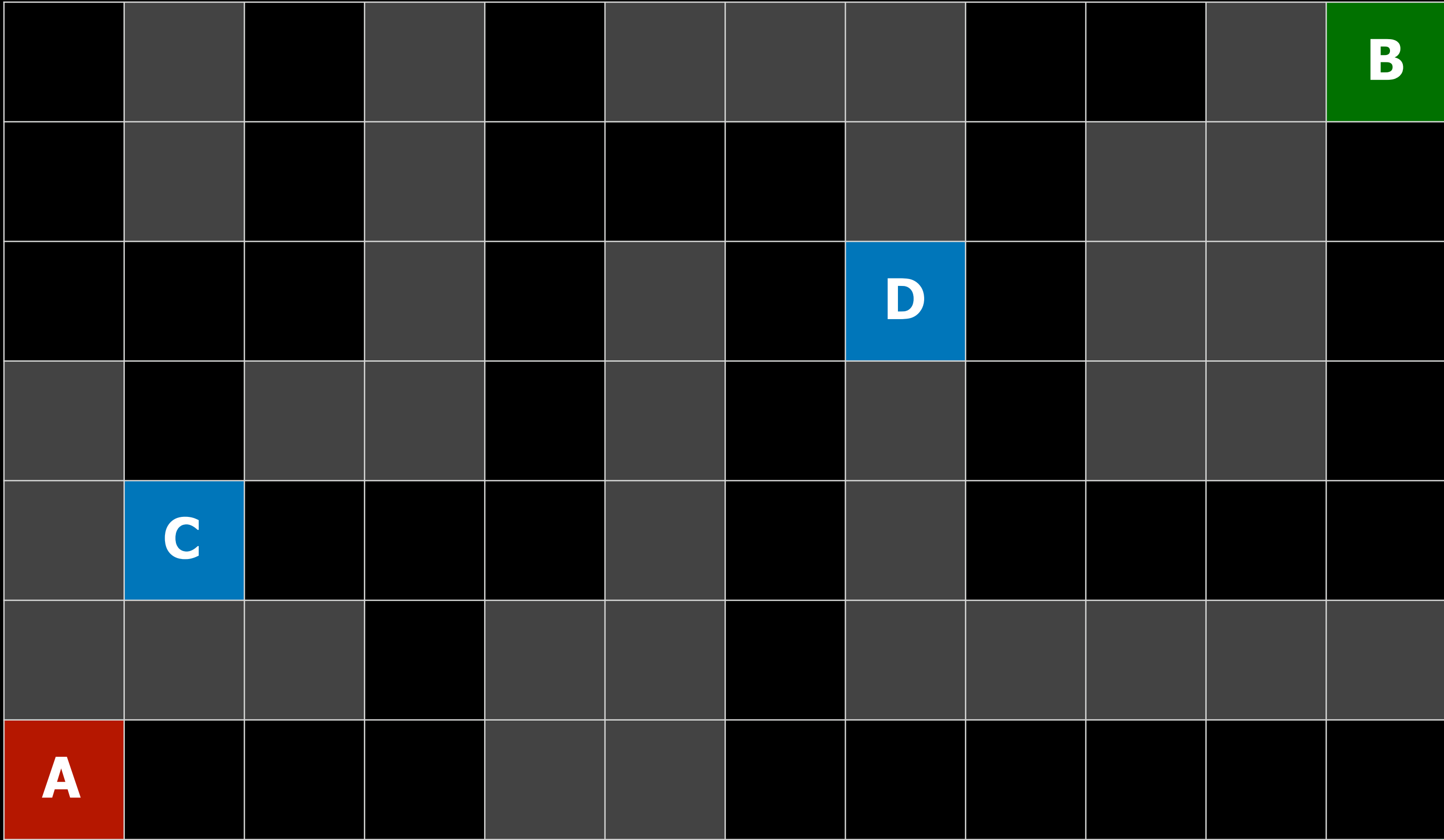
Ventajas y Desventajas del Uso de Heurísticas

- Ventajas:
 - Permiten reducir el tiempo de búsqueda en problemas complejos.
 - Son aplicables a una variedad de problemas (rutas, juegos, planificación).
- Desventajas:
 - No garantizan soluciones óptimas a menos que la heurística sea admisible (nunca sobreestima el costo real).
 - La calidad de los resultados depende de la calidad de la función heurística.

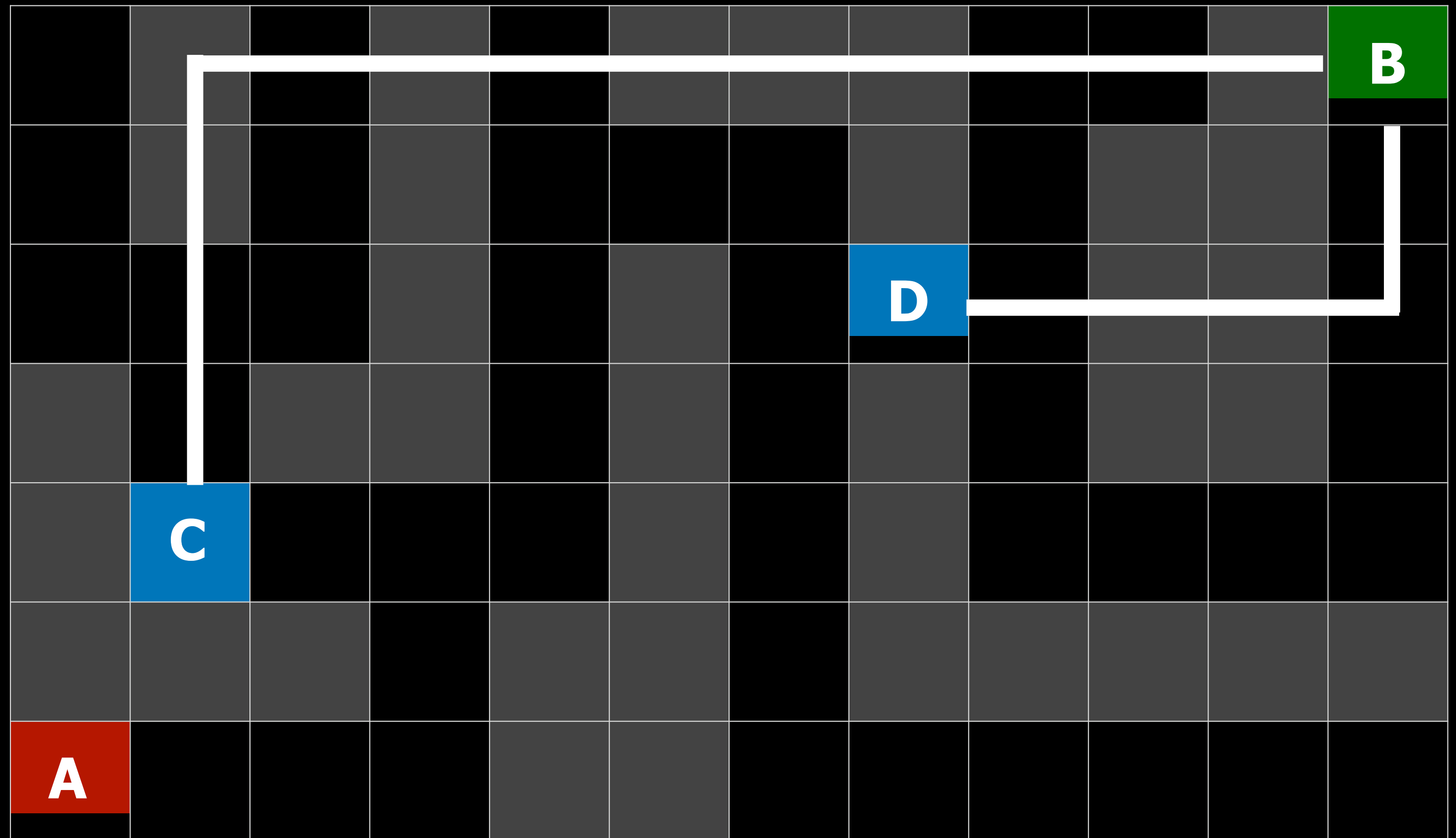
función heurística?



Heuristic function?



Heuristic function? Manhattan distance.



Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Greedy Best-First Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

Greedy Best-First Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

Greedy Best-First Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

A*

search

search algorithm that expands node with

lowest VALOR of $g(n) + h(n)$

$g(n)$ = cost to reach node

$h(n)$ = estimated cost to goal

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	1+16	15	14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	1+16	2+15	14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	9	8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
	14	6+13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
	14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	10	9	8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	9	8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	13+8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	13+8	14+7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	13+8	14+7	15+6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	13+8	14+7	15+6	16+5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	13+8	14+7	15+6	16+5	17+4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	13+8	14+7	15+6	16+5	17+4	18+3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

	11+10	12+9	13+8	14+7	15+6	16+5	17+4	18+3	19+2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

A* Search

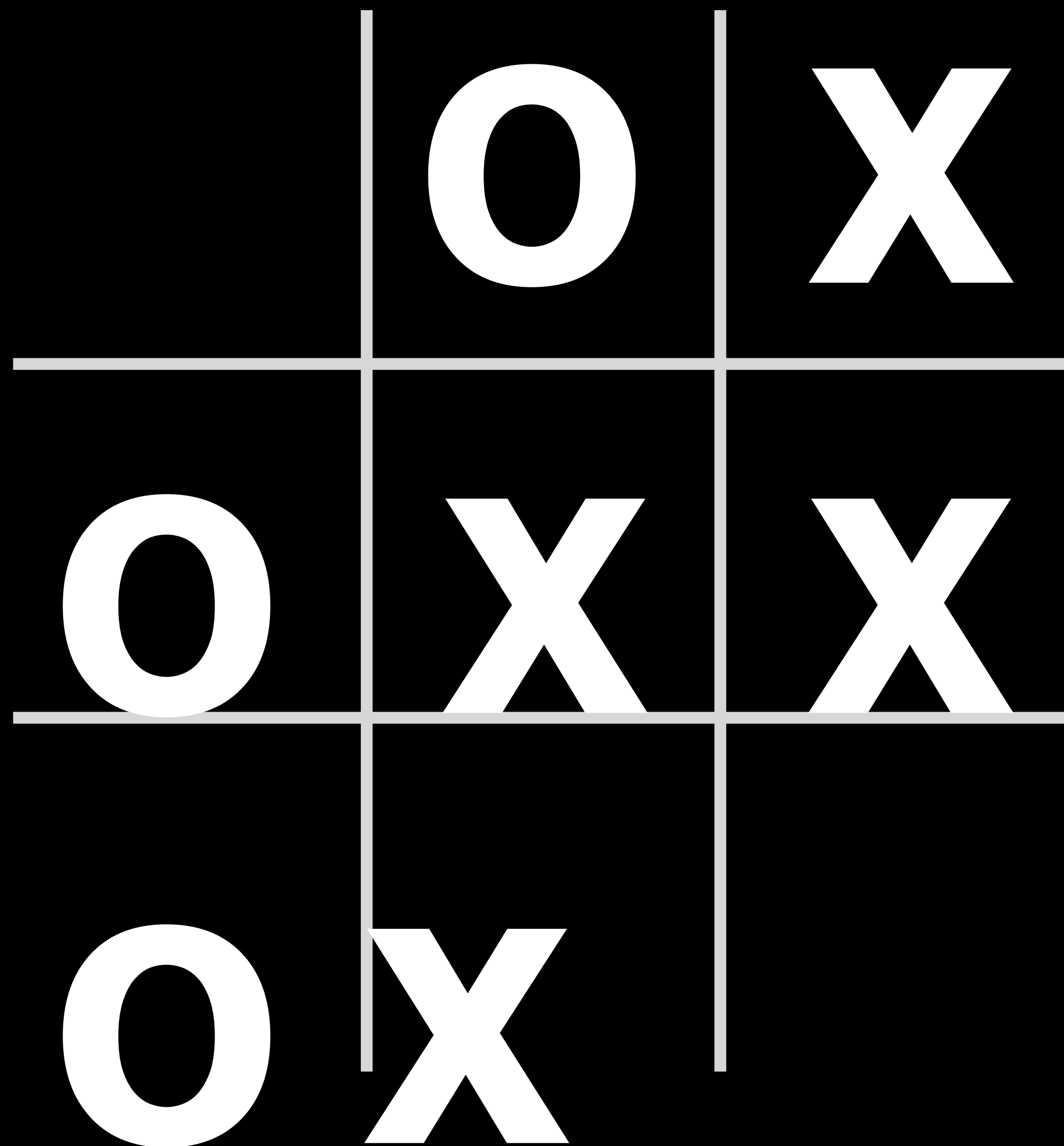
	11+10	12+9	13+8	14+7	15+6	16+5	17+4	18+3	19+2	20+1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

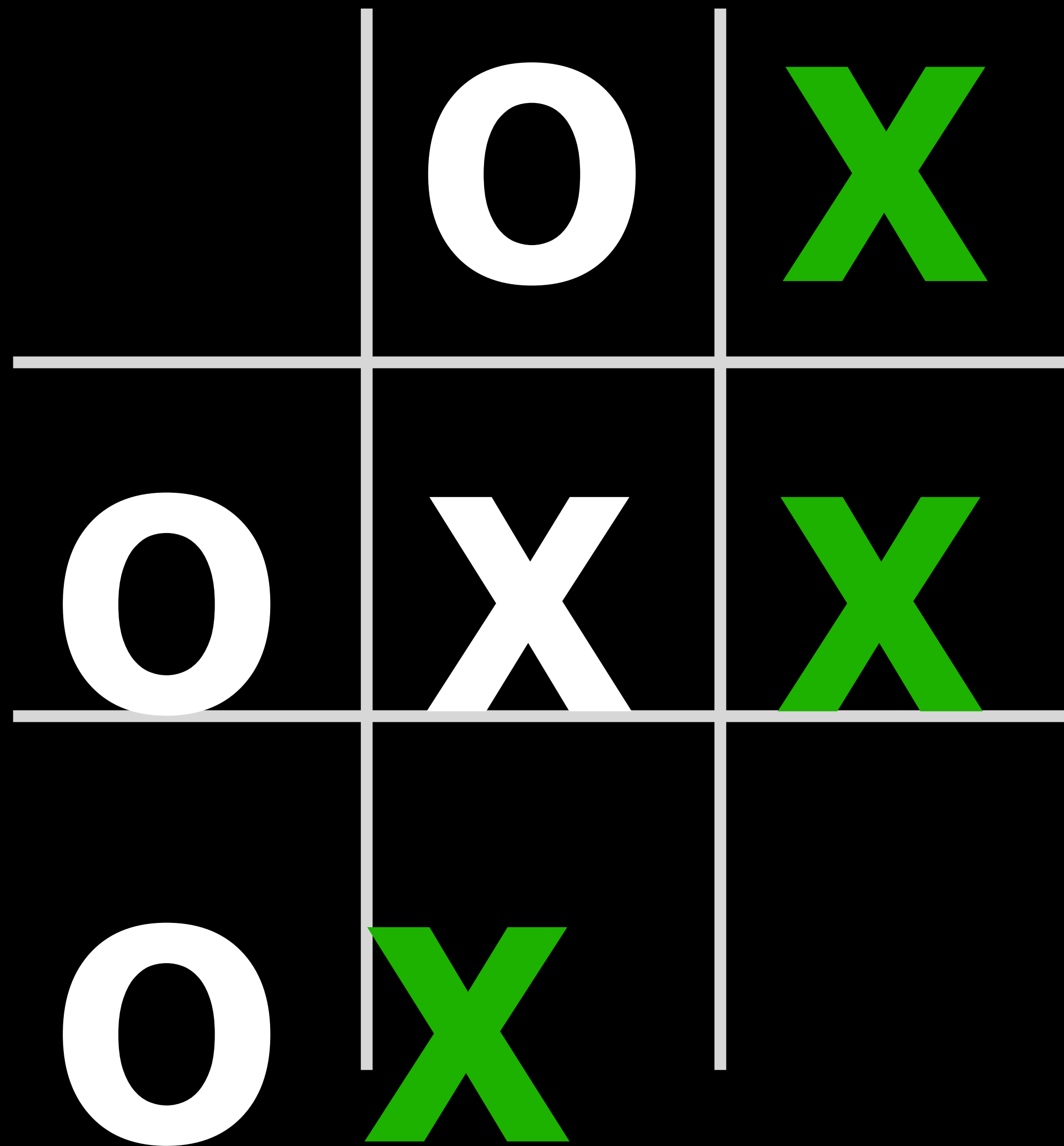
A* búsqueda

Óptima si:

- $h(n)$ es admisible (nunca sobreestima el verdadero costo),
- $h(n)$ es consistente (para cada nodo n y nodo sucesor n' con costo de paso c , $h(n) \leq h(n') + c$)

Búsqueda con situación adversa





Minimax

O	X	X
O	O	
O	X	X

-1

X	O	X
O	O	X
X	X	O

0

O		X
	X	O
X	O	X

1

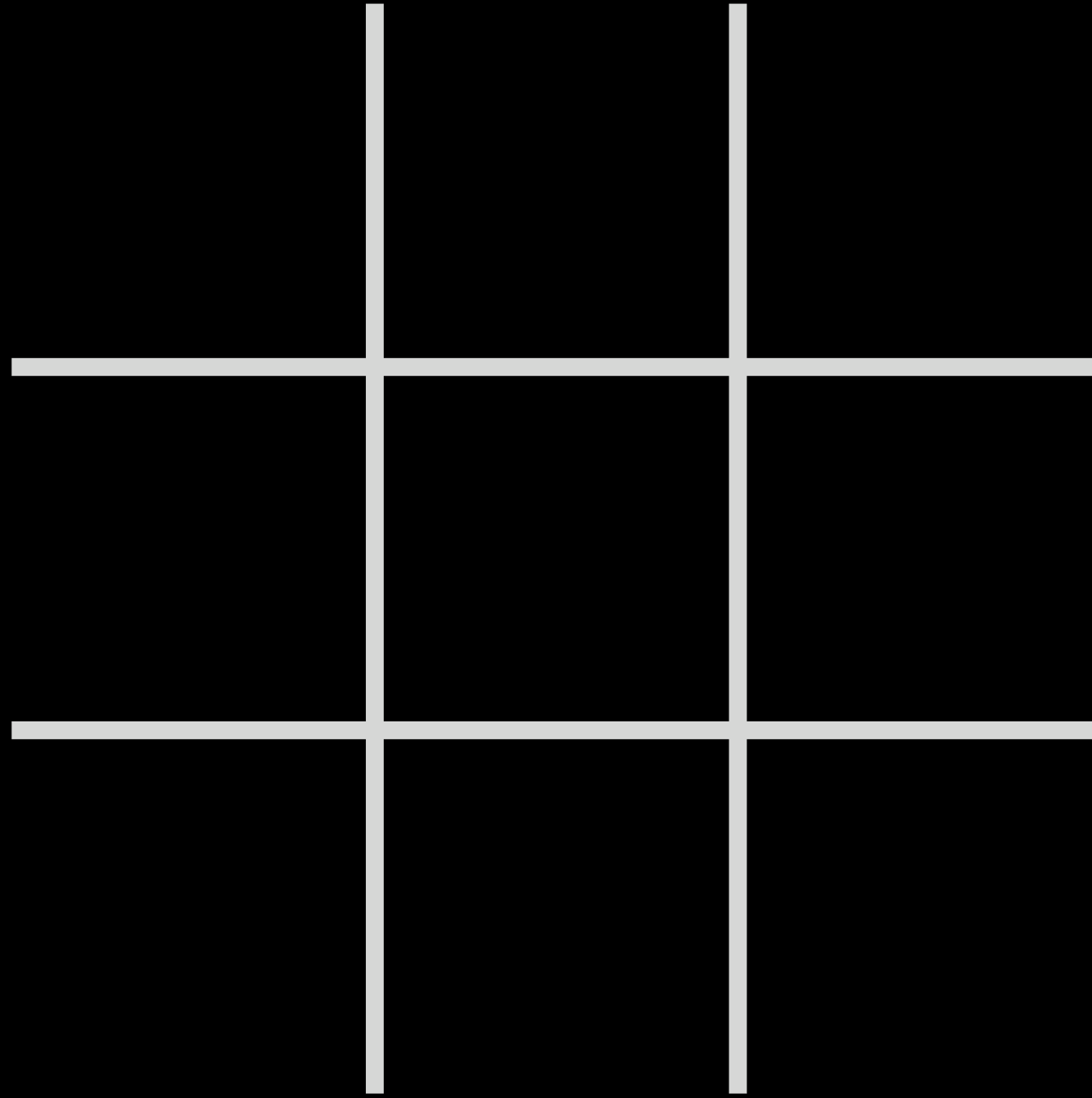
Minimax

- MAX (X) trata de maximizar el puntaje.
- MIN (O) trata de minimizar el puntaje.

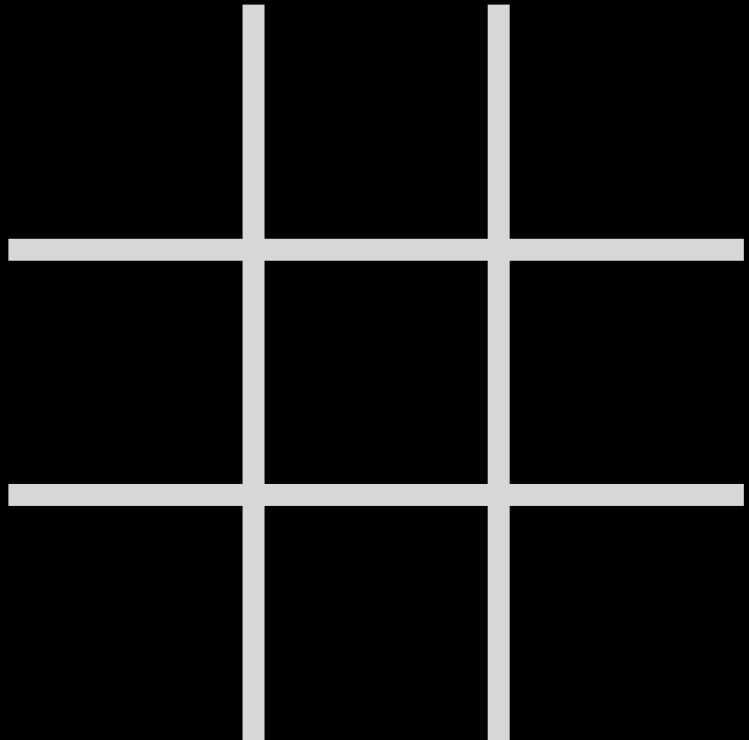
Game

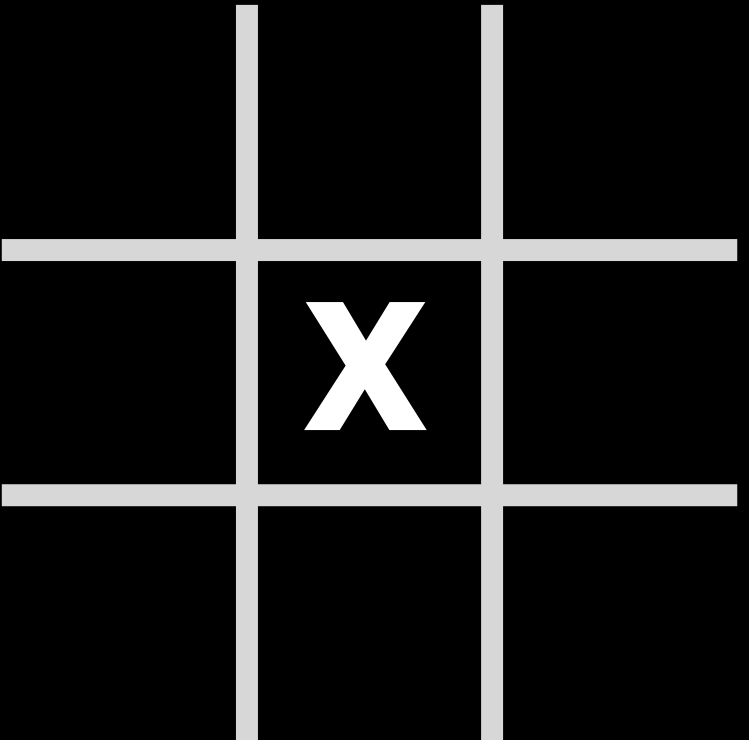
- S_0 : estado inicial
- $\text{JUGADOR}(s)$: retorna qué jugador mueve en el estado s
- $\text{ACCION}(s)$: retorna un movimiento autorizado en el estado s
- $\text{RESULTADO}(s, a)$: retorna el estado después de una acción a tomada en un estado s
- $\text{TERMINAL}(s)$: chequee si el estado s es un estado terminal
- $\text{UTILIDAD}(s)$: valor numérico final para un estado s

Estado
Inicial



Jugador(s)

Jugador() = **x**

Jugador() = **o**

ACCIONES(s)

$$\text{ACCIONES}\left(\begin{array}{c|c|c} & \mathbf{x} & \mathbf{o} \\ \hline \mathbf{o} & \mathbf{x} & \mathbf{x} \\ \hline \mathbf{x} & & \mathbf{o} \end{array}\right) = \left\{ \begin{array}{c|c|c} \mathbf{o} & & \\ \hline & & \\ \hline & & \end{array}, \begin{array}{c|c|c} & & \\ \hline & & \\ \hline & \mathbf{o} & \end{array} \right\}$$

RESULT(s, a)

$$\text{RESULT}\left(\begin{array}{c|c|c} & \mathbf{x} & \mathbf{o} \\ \hline \mathbf{o} & \mathbf{x} & \mathbf{x} \\ \hline \mathbf{x} & & \mathbf{o} \end{array}, \begin{array}{c|c|c} \mathbf{o} & & \\ \hline \hline \hline \end{array}\right) = \begin{array}{c|c|c} \mathbf{o} & \mathbf{x} & \mathbf{o} \\ \hline \mathbf{o} & \mathbf{x} & \mathbf{x} \\ \hline \mathbf{x} & & \mathbf{o} \end{array}$$

TERMINAL(s)

TERMINAL(

o		
o	x	
x	o	x

) = false

TERMINAL(

o	x	x
o	x	
x	o	x

) = true

UTILIDAD(s)

$$\text{UTILIDAD}\left(\begin{array}{c|c|c} \mathbf{o} & \mathbf{x} & \mathbf{x} \\ \hline \mathbf{o} & \mathbf{x} & \\ \hline \mathbf{x} & \mathbf{o} & \mathbf{x} \end{array}\right) = 1$$

$$\text{UTILIDAD}\left(\begin{array}{c|c|c} \mathbf{o} & \mathbf{x} & \mathbf{x} \\ \hline \mathbf{x} & \mathbf{o} & \\ \hline \mathbf{o} & \mathbf{x} & \mathbf{o} \end{array}\right) = -1$$

O	X	O
O	X	X
X	X	O

VALOR:

1

$$\text{Jugador}(s) = \text{O}$$

MIN-VALOR

:

	X	O
O	X	X
X		O

\emptyset

MAX-VALOR:

1

O	X	O
O	X	X
X		O



VALOR:

1

O	X	O
O	X	X
X	X	O

MAX-VALOR:

\emptyset

	X	O
O	X	X
X	O	O



VALOR:

\emptyset

X	X	O
O	X	X
X	O	O

$$\text{Jugador}(s) = \text{O}$$

MIN-VALOR:

\emptyset

	X	O
O	X	X
X		O

MAX-VALOR:

1

O	X	O
O	X	X
X		O



VALOR:

1

O	X	O
O	X	X
X	X	O

MAX-VALOR:

\emptyset

	X	O
O	X	X
X	O	O



VALOR:

\emptyset

X	X	O
O	X	X
X	O	O

$\text{PLAYER}(s) = X$

MAX-VALO
 R: 1

	X	O
O	X	
X		O

MIN-VALO
 R:

	X	O
O	X	X
X		O

\emptyset

MAX-VALO
 R: 1

O	X	O
O	X	X
X		O



VALO
 R: 1

O	X	O
O	X	X
X	X	O

MAX-VALO
 R: \emptyset

	X	O
O	X	X
X	O	O



VALO
 R: \emptyset

X	X	O
O	X	X
X	O	O

MIN-VALO
 R:

X	X	O
O	X	
X		O

-1

VALO
 R: -1

X	X	O
O	X	O
X		O

MAX-VALO
 R: \emptyset

X	X	O
O	X	
X	O	O

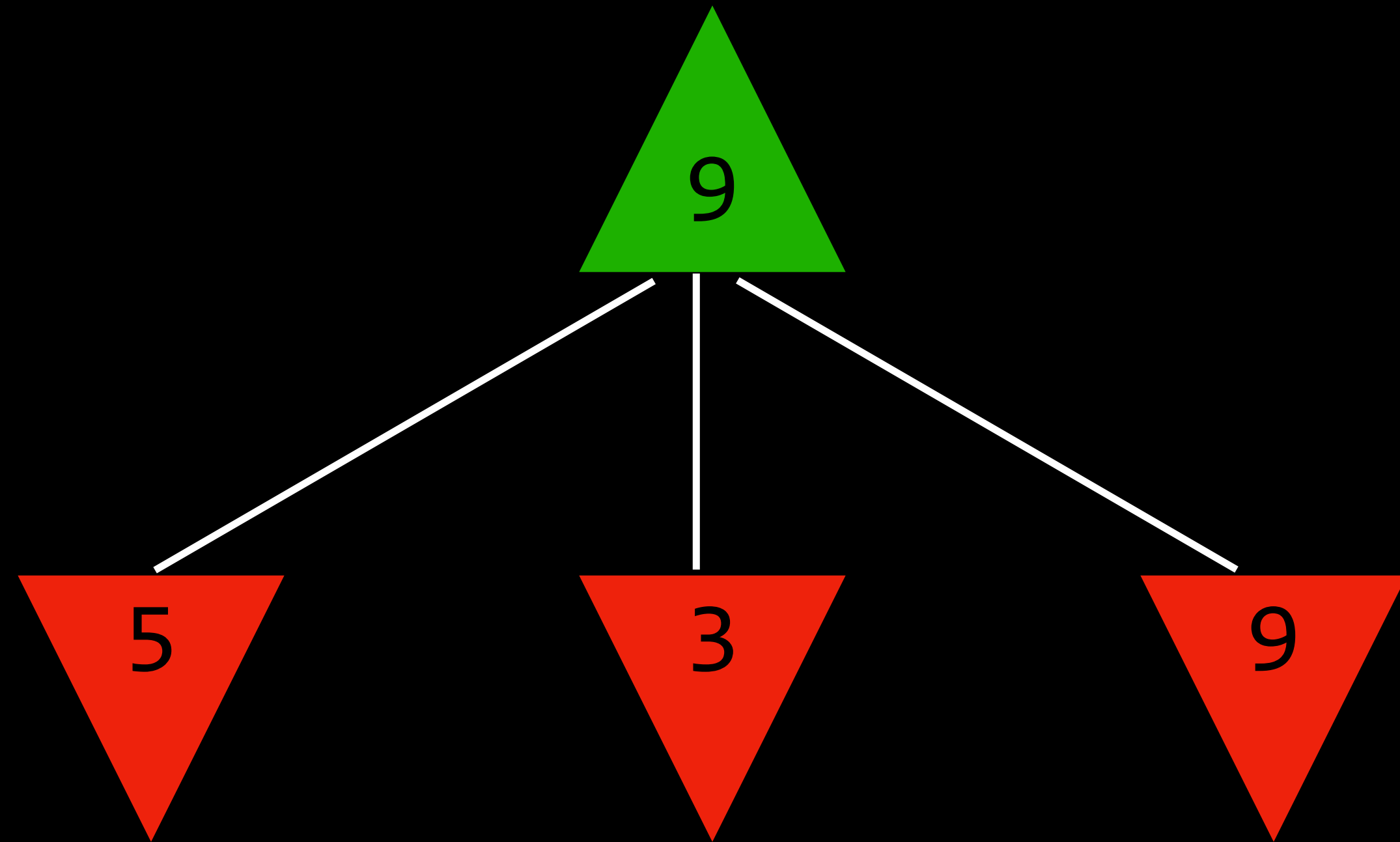


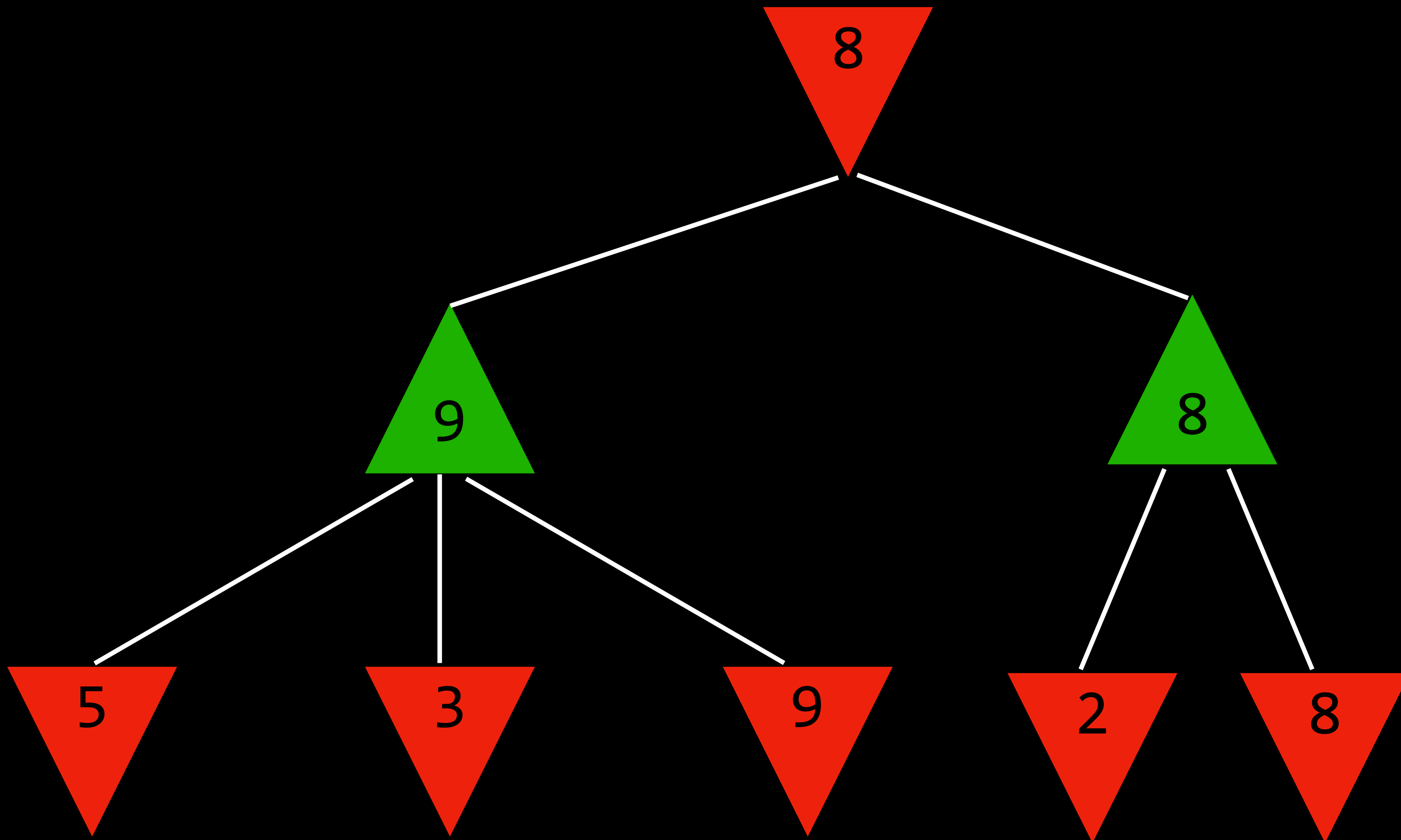
VALO
 R: \emptyset

X	X	O
O	X	X
X	O	O

VALO
 R: 1

X	O	
O	X	
X	X	O





Minimax

- Dado un estado s :
 - MAX toma una acción a en el conjunto $\text{ACCIONES}(s)$ que produce el valor mas alto VALOR de $\text{MIN-VALOR}(\text{RESULTADOS}(s, a))$
 - MIN elige una acción a ien el conjunto $\text{Acciones}(s)$ que produce el valor mas pequeñoequeuo VALOR de $\text{MAX-VALOR}(\text{RESULT}(s, a))$

Minimax

function

MAX-VALOR(*state*): if

TERMINAL(*state*):

return UTILITY(*state*)

$v = -\infty$

for *action* in ACTIONS(*state*):

$v = \text{MAX}(v, \text{MIN-VALOR}(\text{RESULT}(\text{state},$
action))) return v

Minimax

function

MIN-VALOR(*state*): if

TERMINAL(*state*):

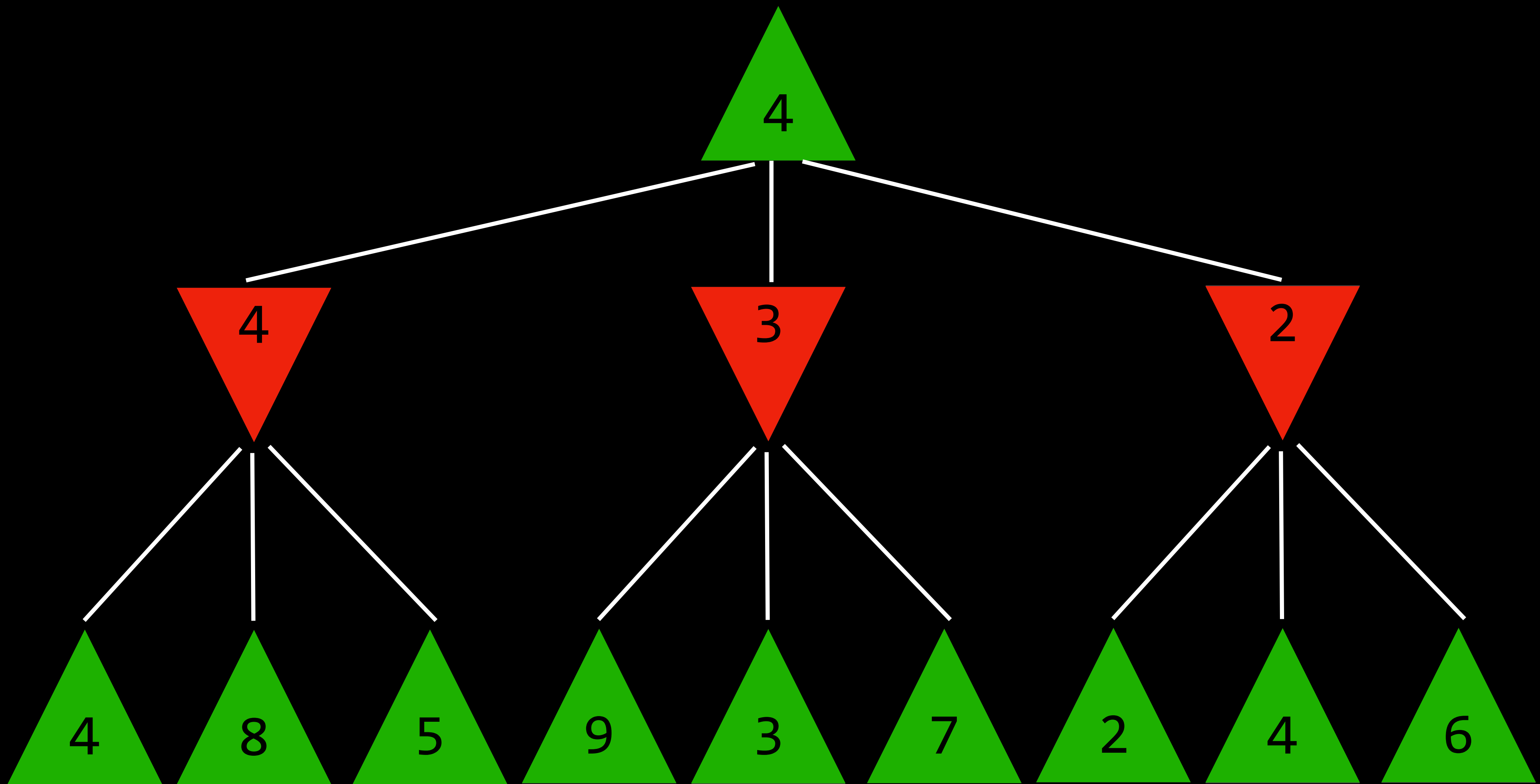
return UTILITY(*state*)

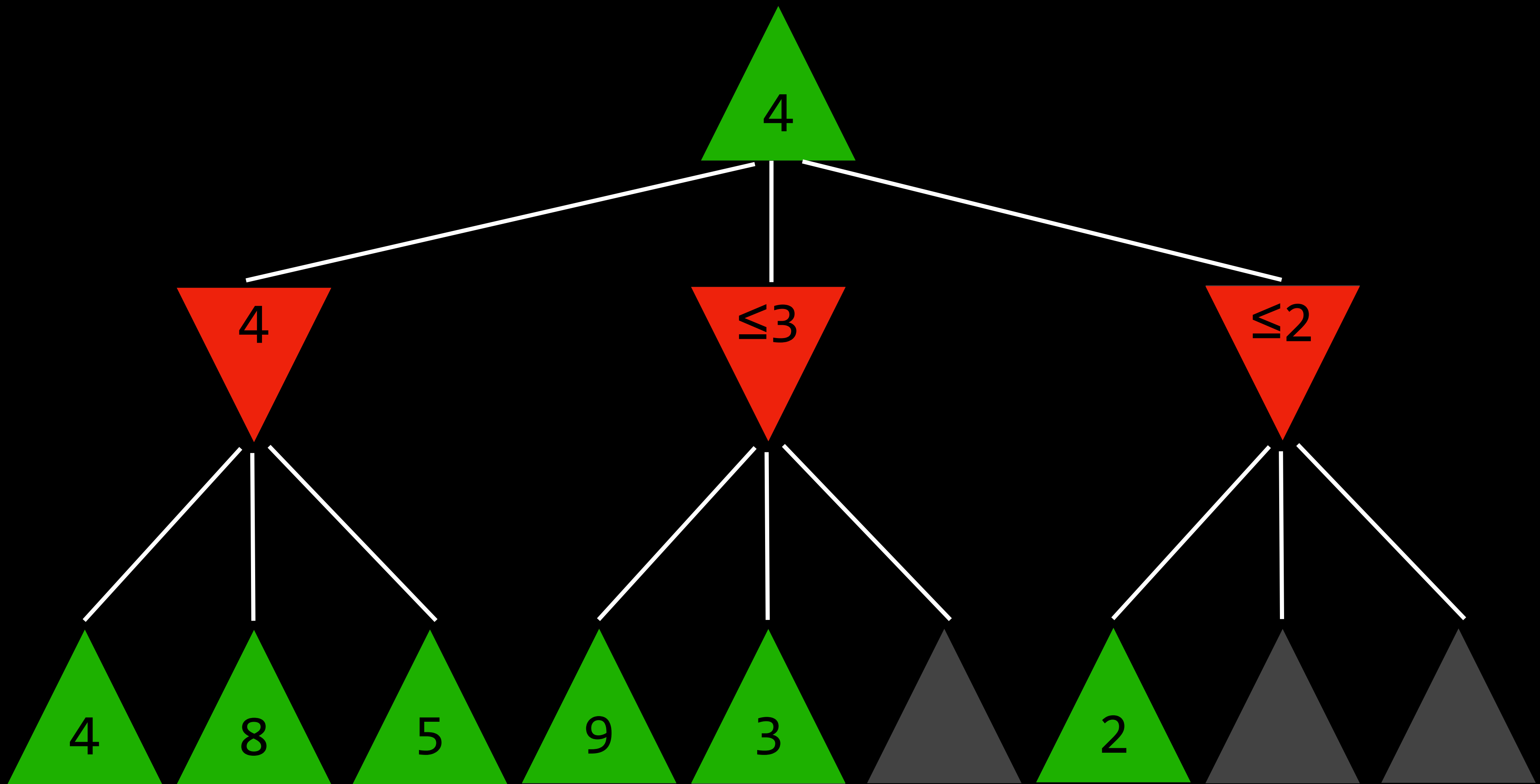
$v = \infty$

for *action* in ACTIONS(*state*):

$v = \text{MIN}(v, \text{MAX-VALOR}(\text{RESULT}(\text{state},$
action))) return v

Optimizations





Alpha-Beta Pruning

255,168

total possible Tic-Tac-Toe games

288,000,000,000

total possible chess games
after four moves each

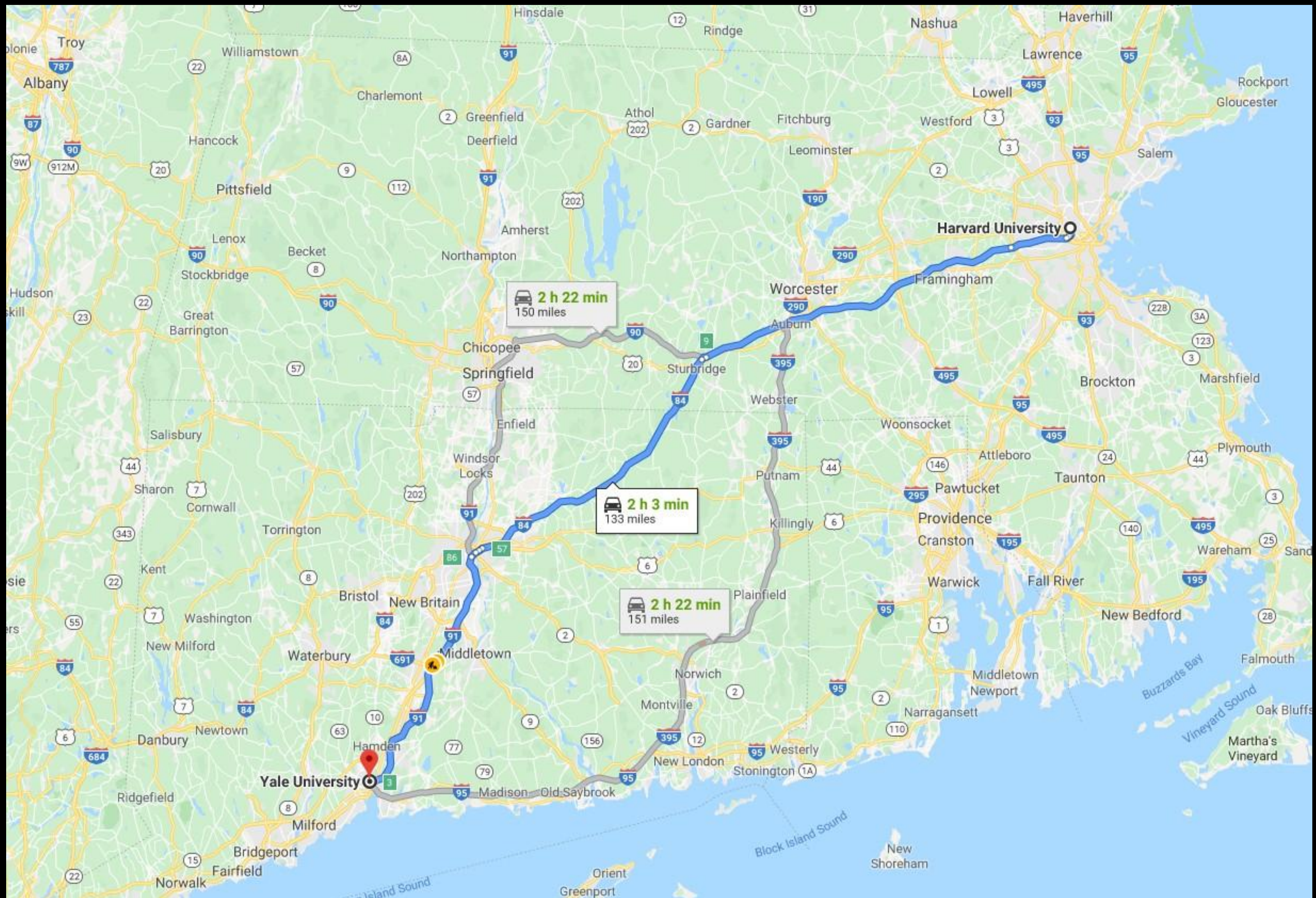
10²⁹⁰⁰⁰

total possible chess games
(lower bound)

Depth-Limited Minimax

evaluation function

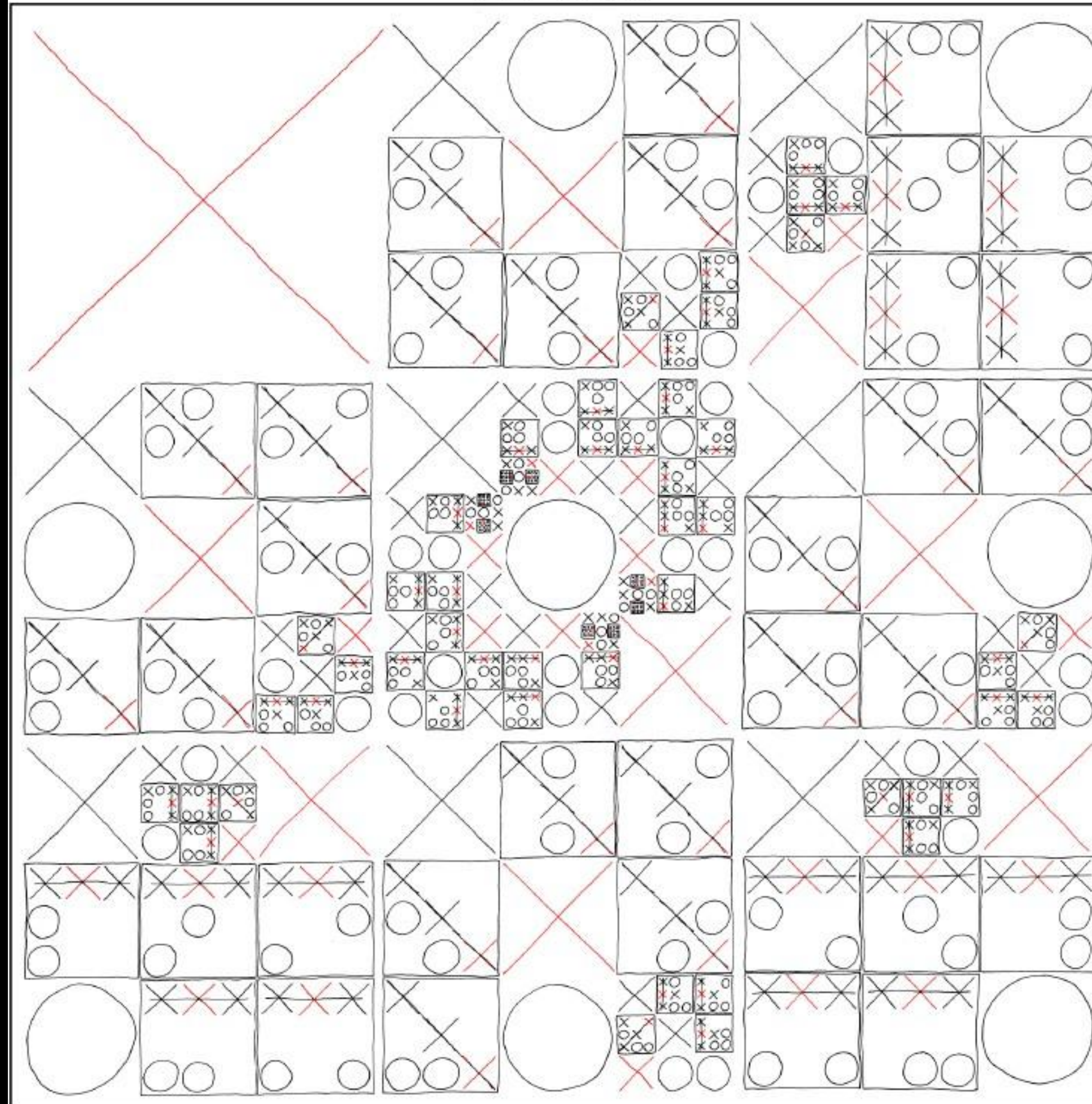
function that estimates the expected utility of
the game from a given state



COMPLETE MAP OF OPTIMAL TIC-TAC-TOE MOVES

YOUR MOVE IS GIVEN BY THE POSITION OF THE LARGEST RED SYMBOL ON THE GRID. WHEN YOUR OPPONENT PICKS A MOVE, ZOOM IN ON THE REGION OF THE GRID WHERE THEY WENT. REPEAT.

MAP FOR X:



Busqueda