

03-29-2023

CS051A

INTRO TO COMPUTER SCIENCE WITH TOPICS IN AI

17: Search



Alexandra Papoutsaki

she/her/hers

Lectures



Zilong Ye

he/him/his

Labs

Lecture 17: Search

- ▶ Search

What is AI?

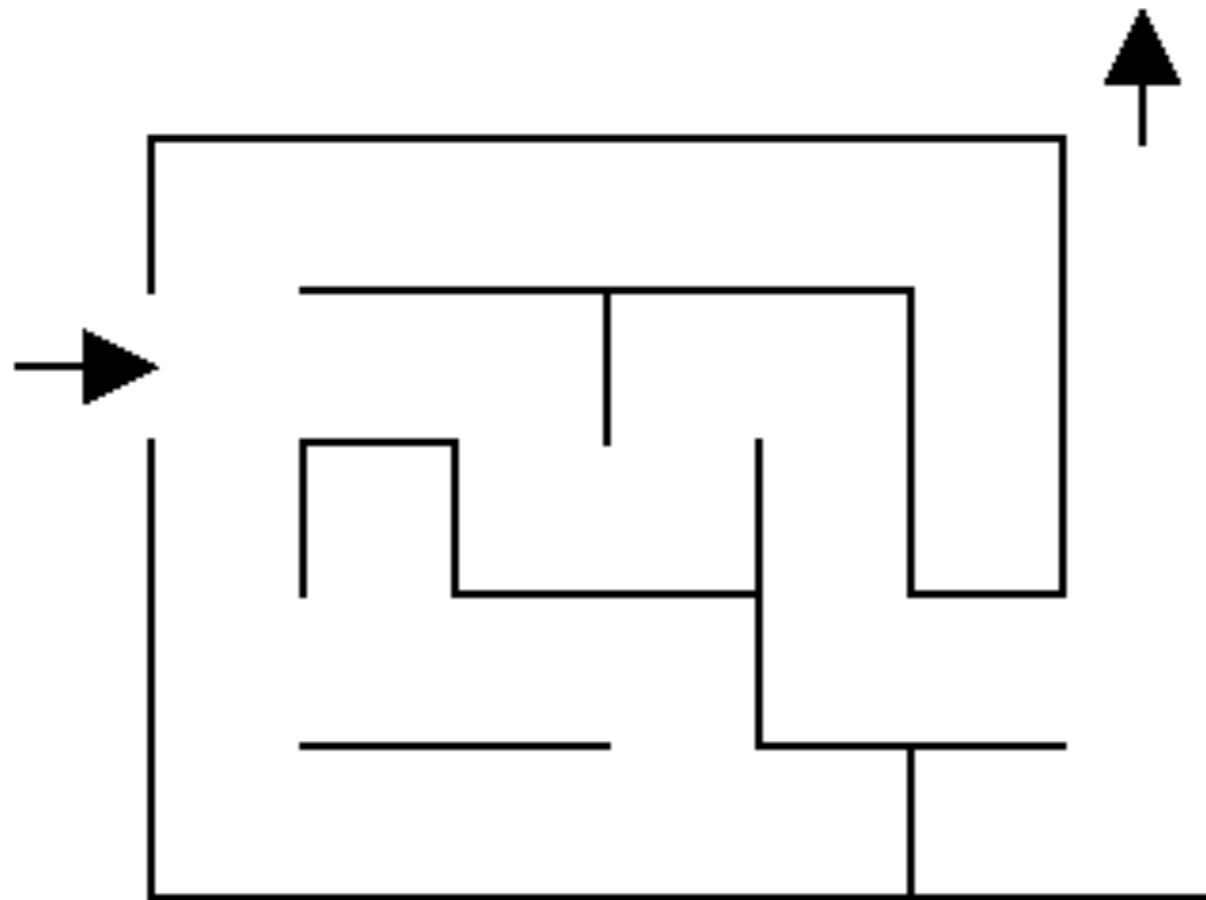
<p>Think like a human Cognitive Modeling</p>	<p>Think rationally Logic-based Systems</p>
<p>Act like a human Turing Test</p>	<p>Act rationally Rational Agents</p>

What is AI?

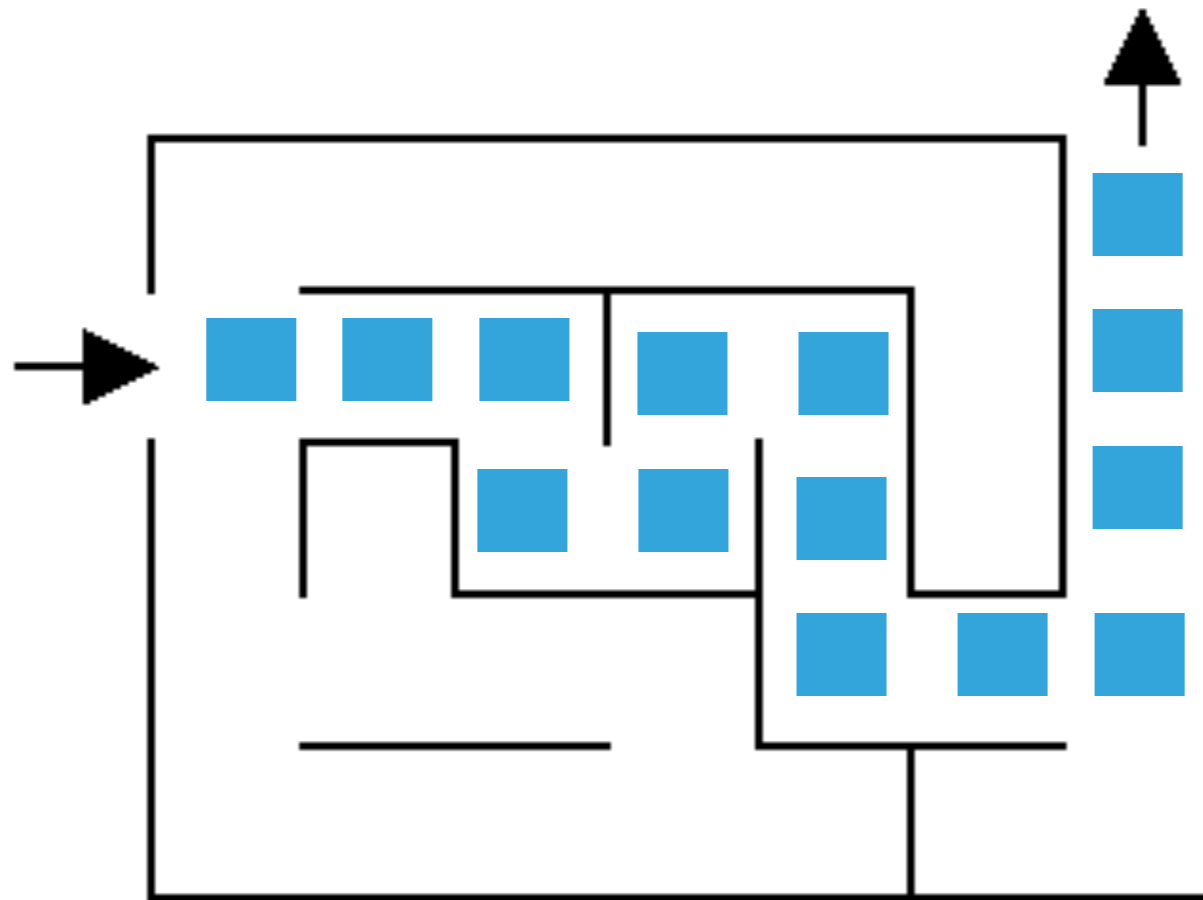
Think like a human Cognitive Modeling	Think rationally Logic-based Systems
Act like a human Turing Test	Act rationally Rational Agents

Next couple of weeks

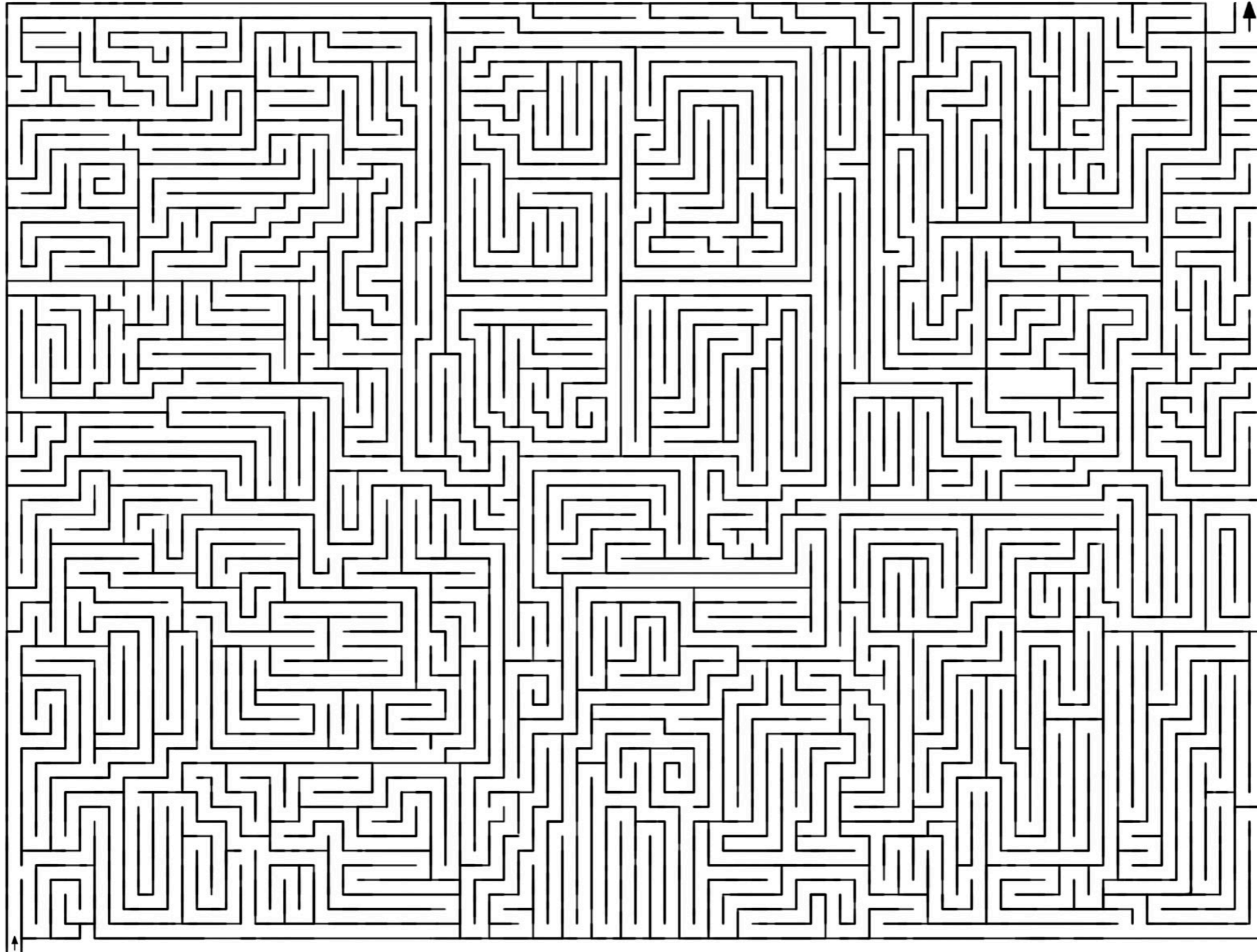
Solve the maze!



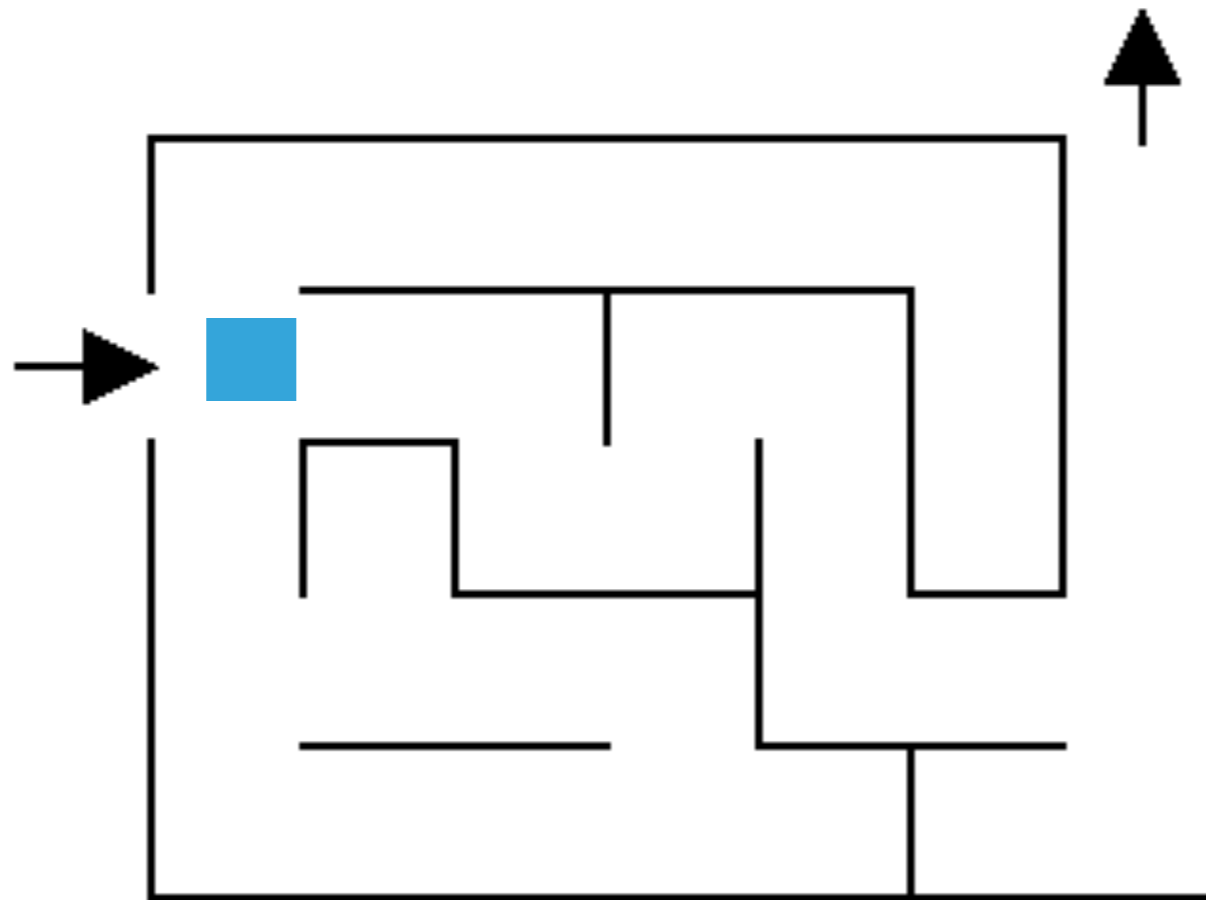
Solve the maze!



Solve the maze!

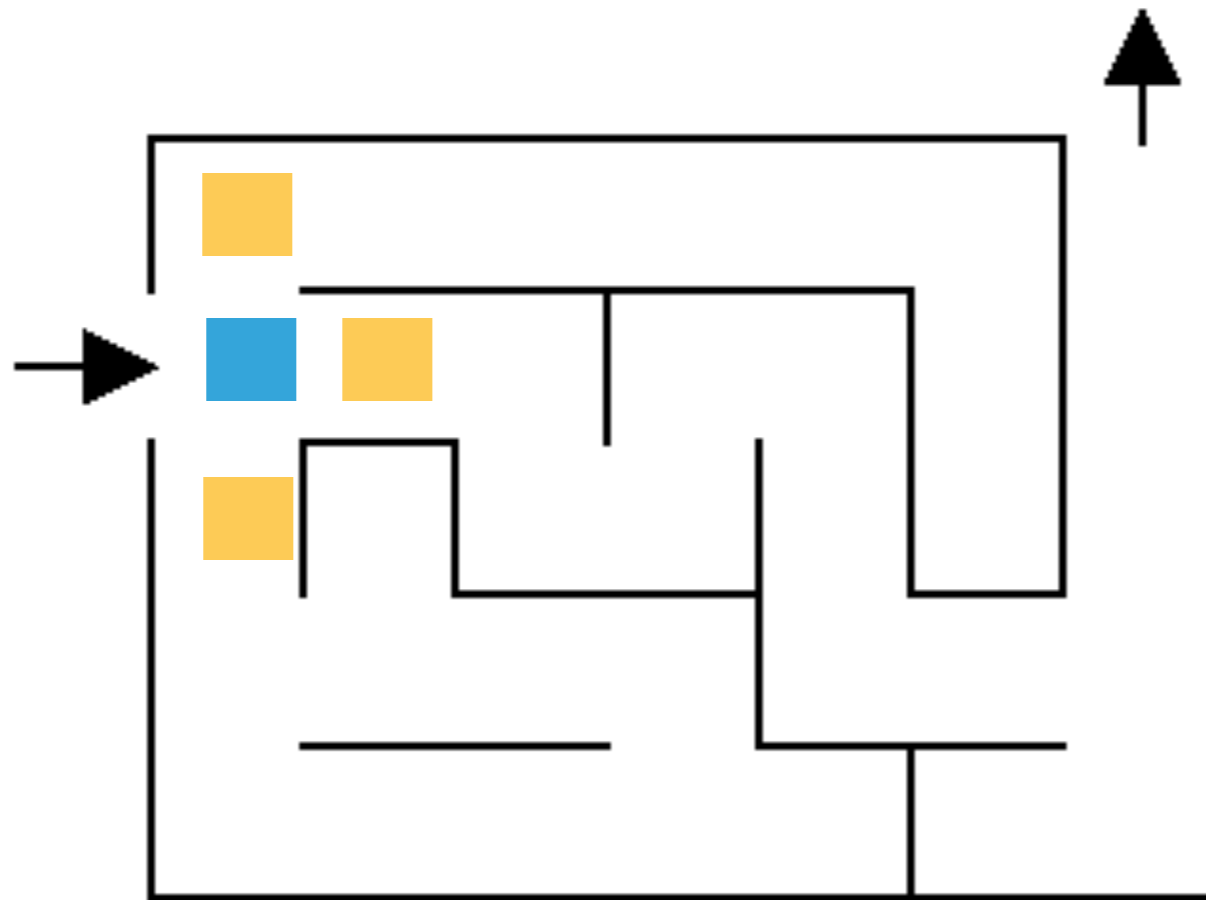


One approach



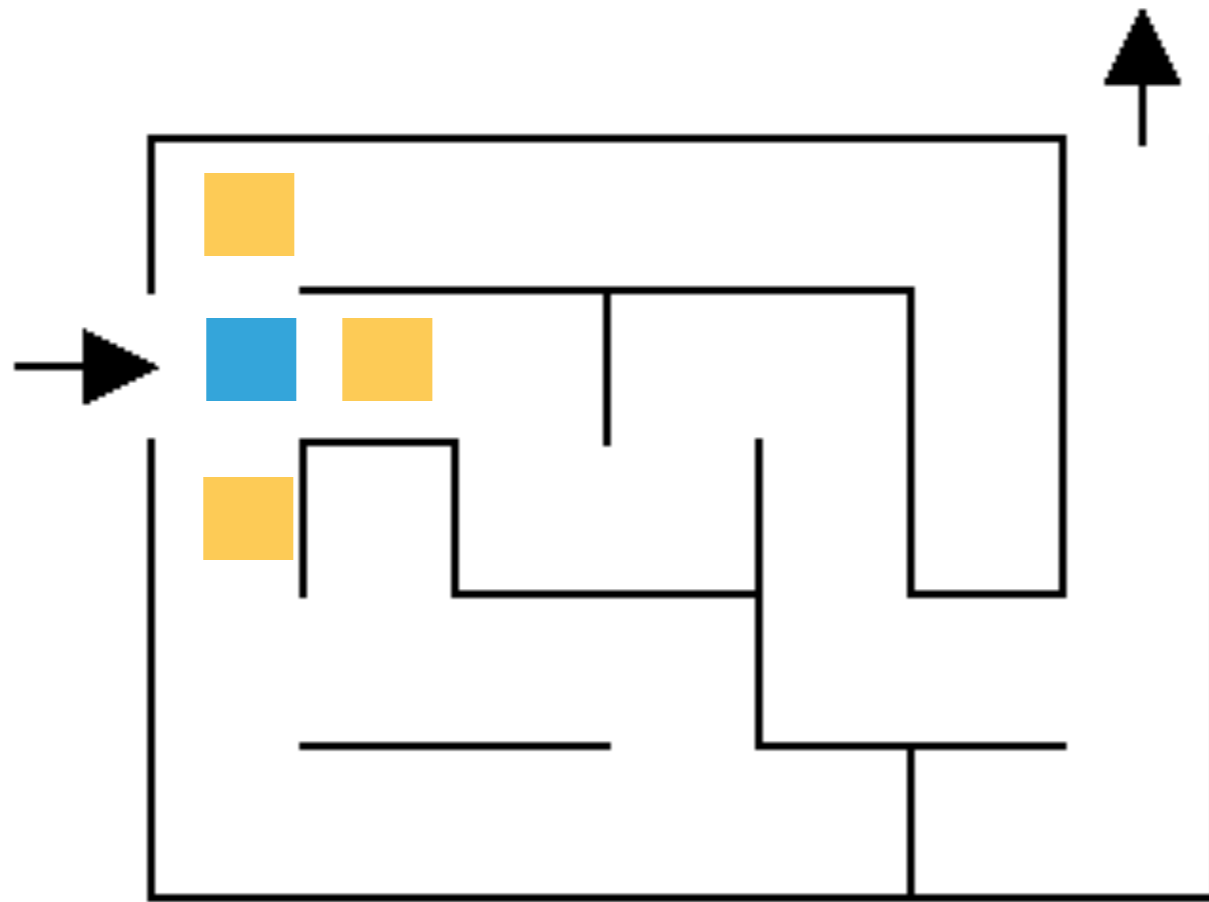
What now?

One approach



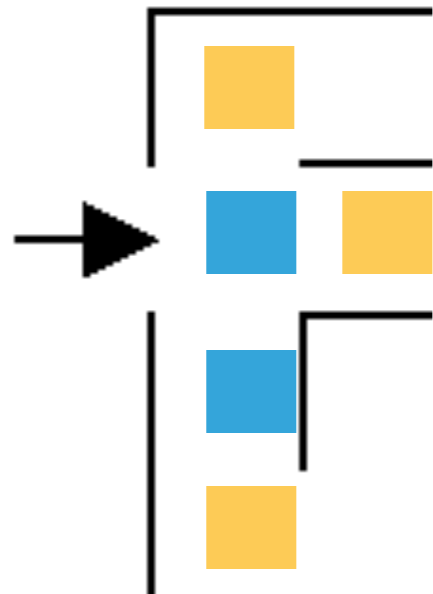
Three choices

One approach



Pick one!

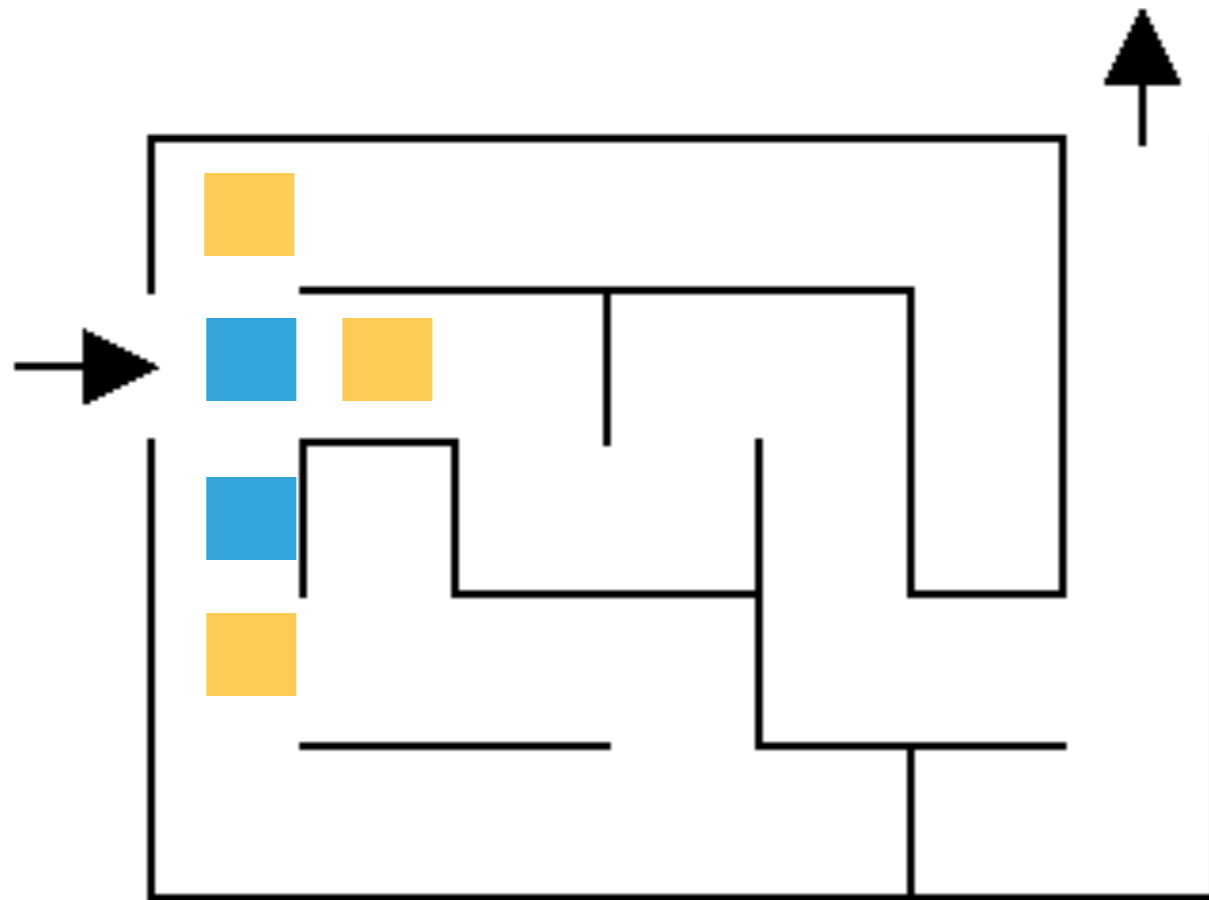
One approach



Still three options!

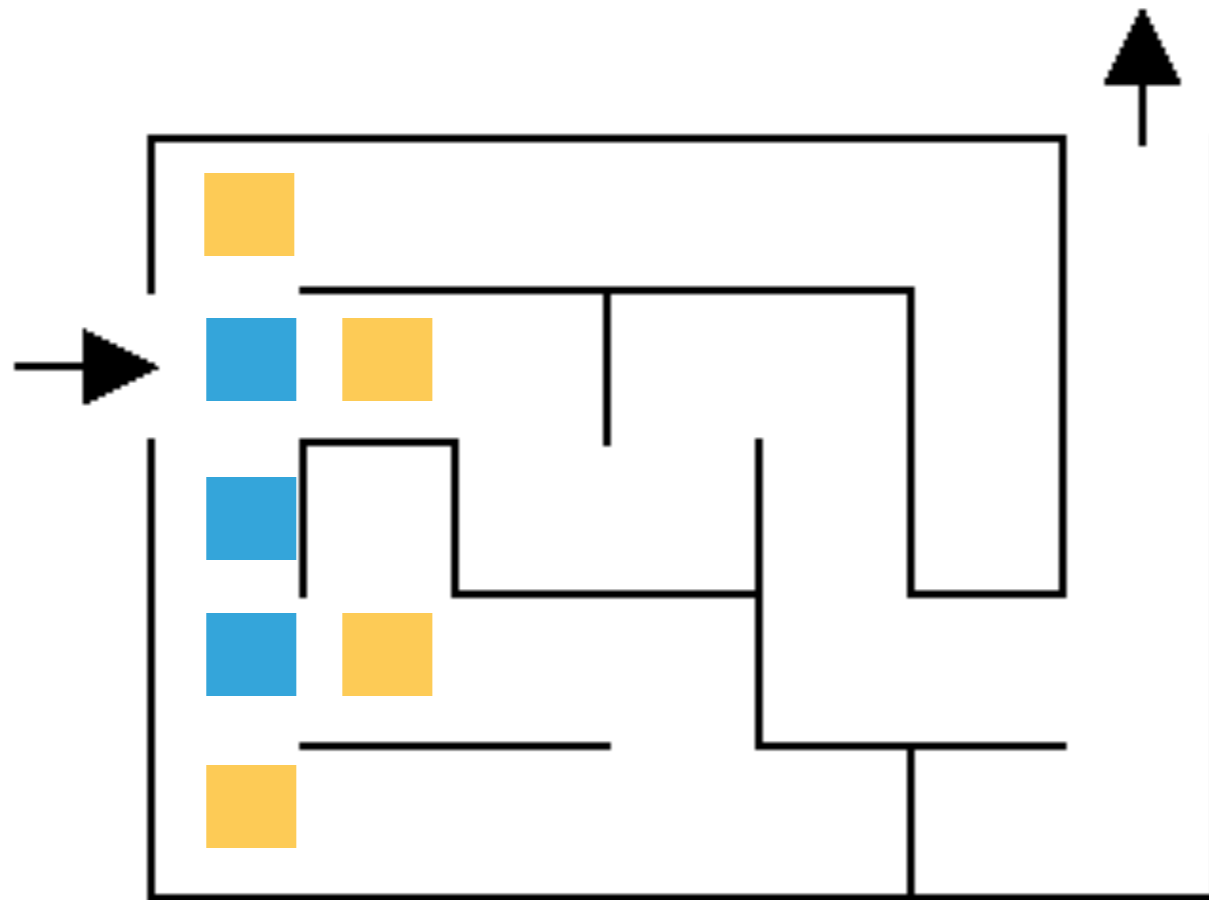
Which would you explore/pick?

One approach



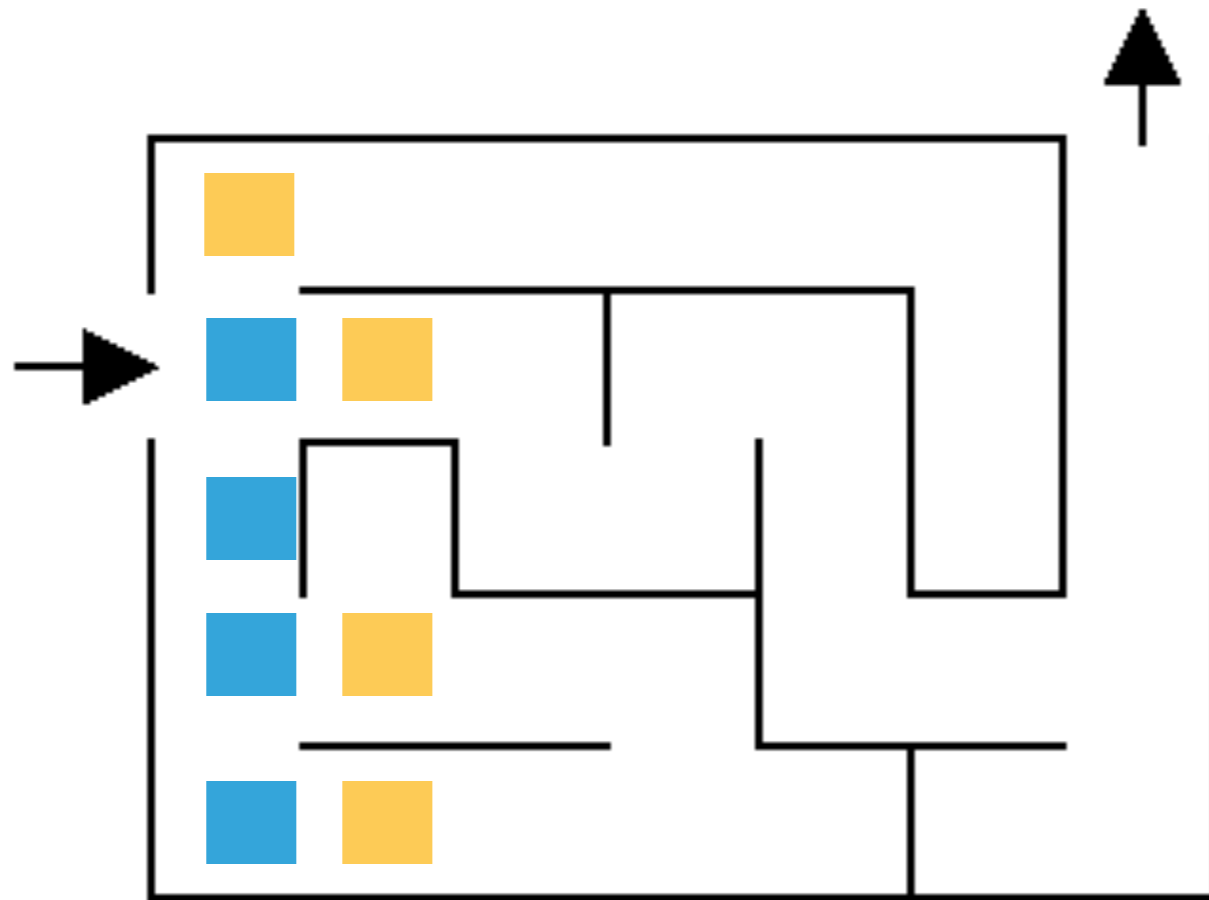
Most people go down a single path
until they realize that it's wrong

One approach



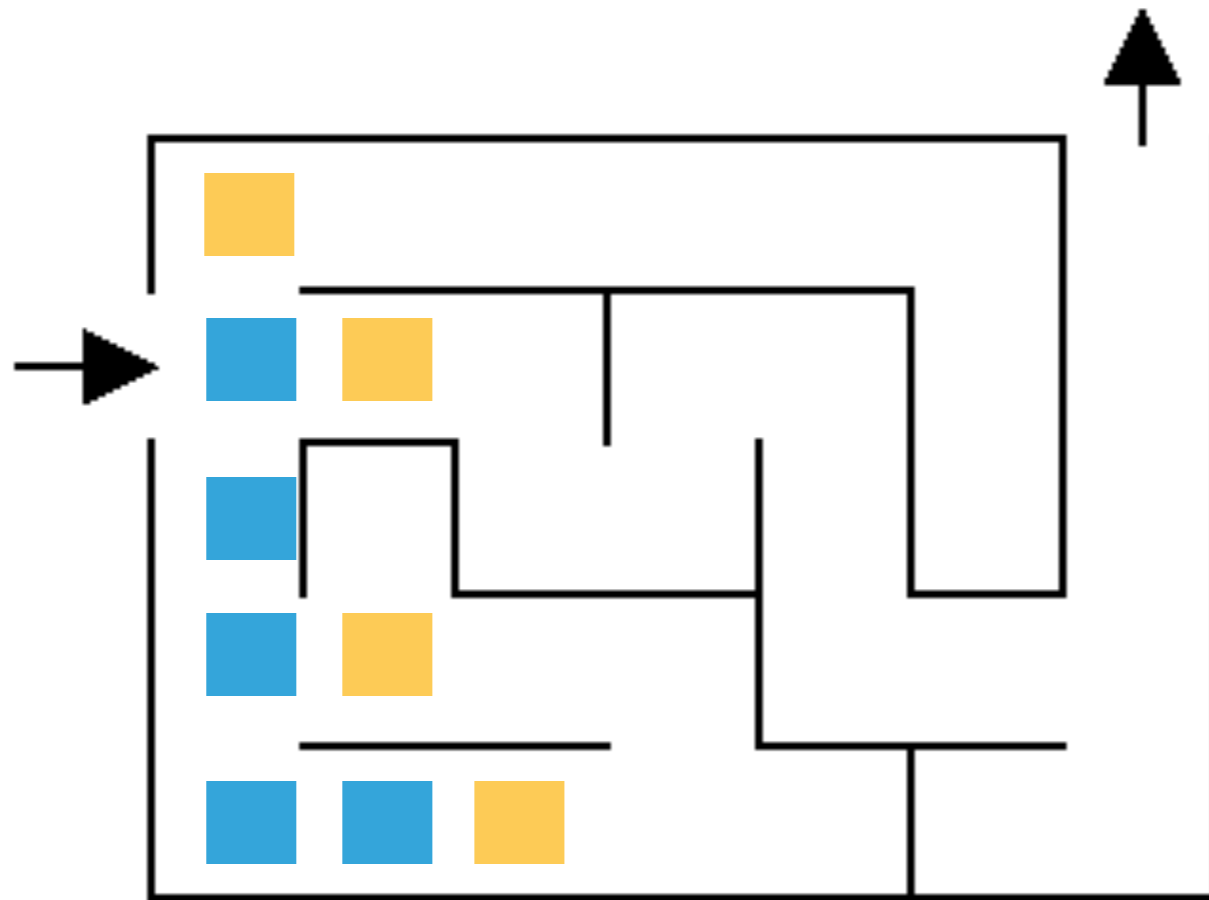
Keep exploring

One approach



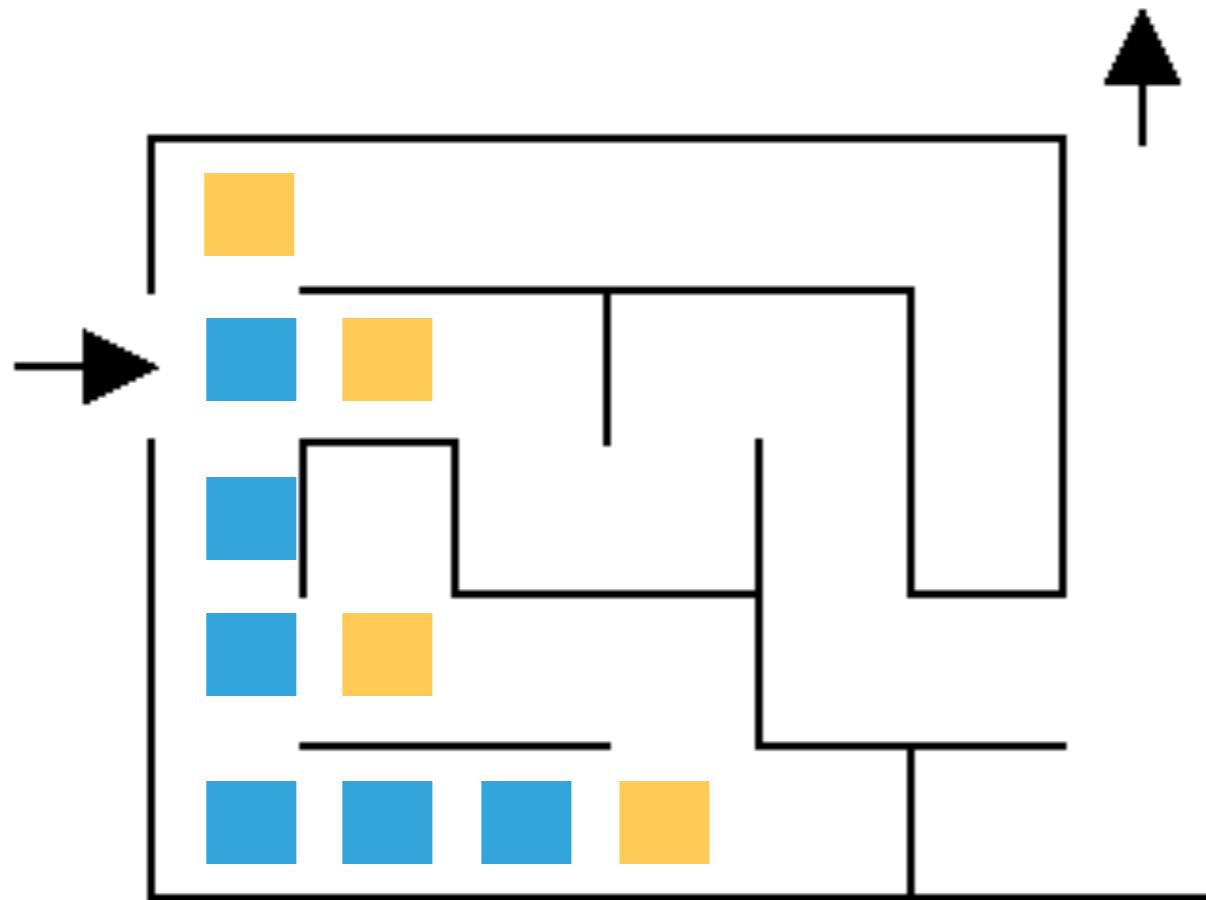
Keep exploring

One approach



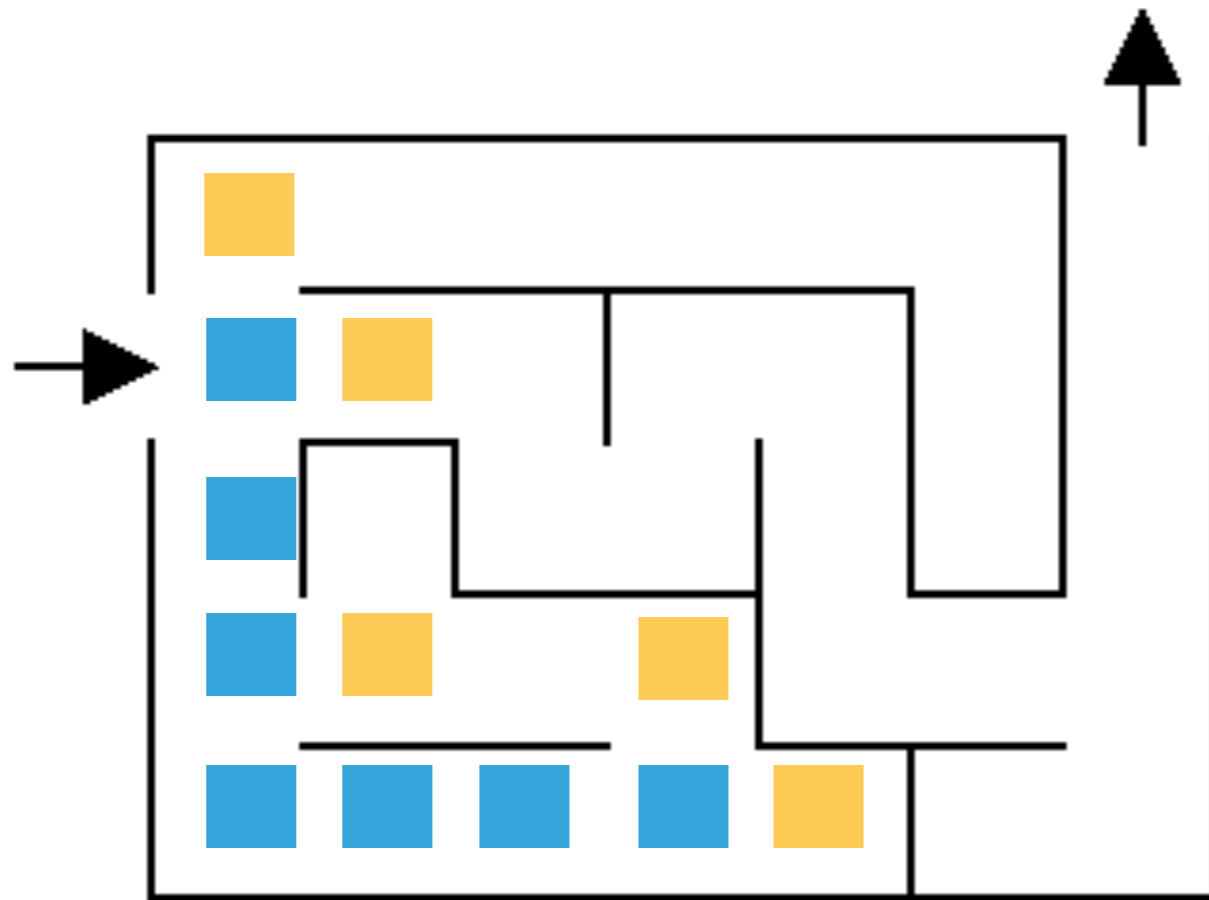
Keep exploring

One approach



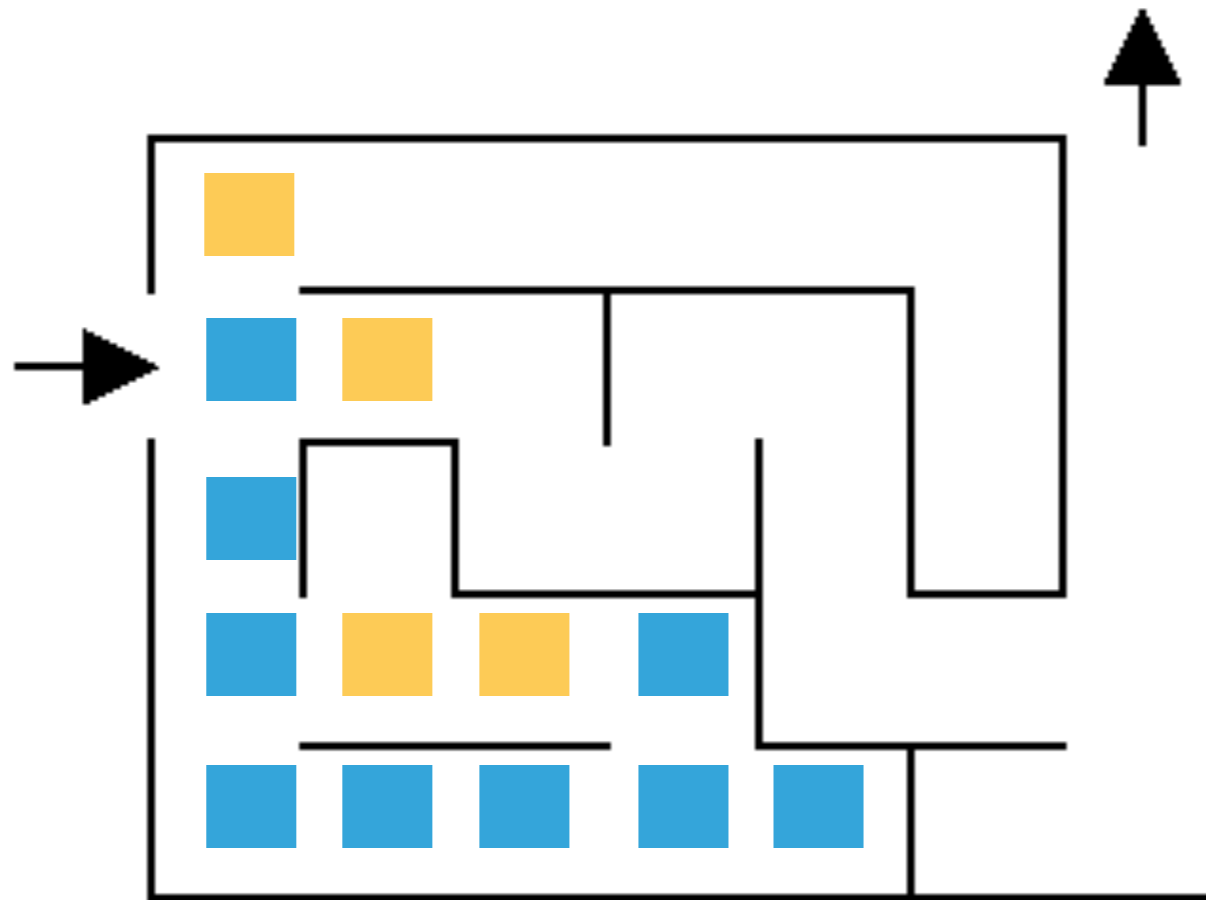
Keep exploring

One approach



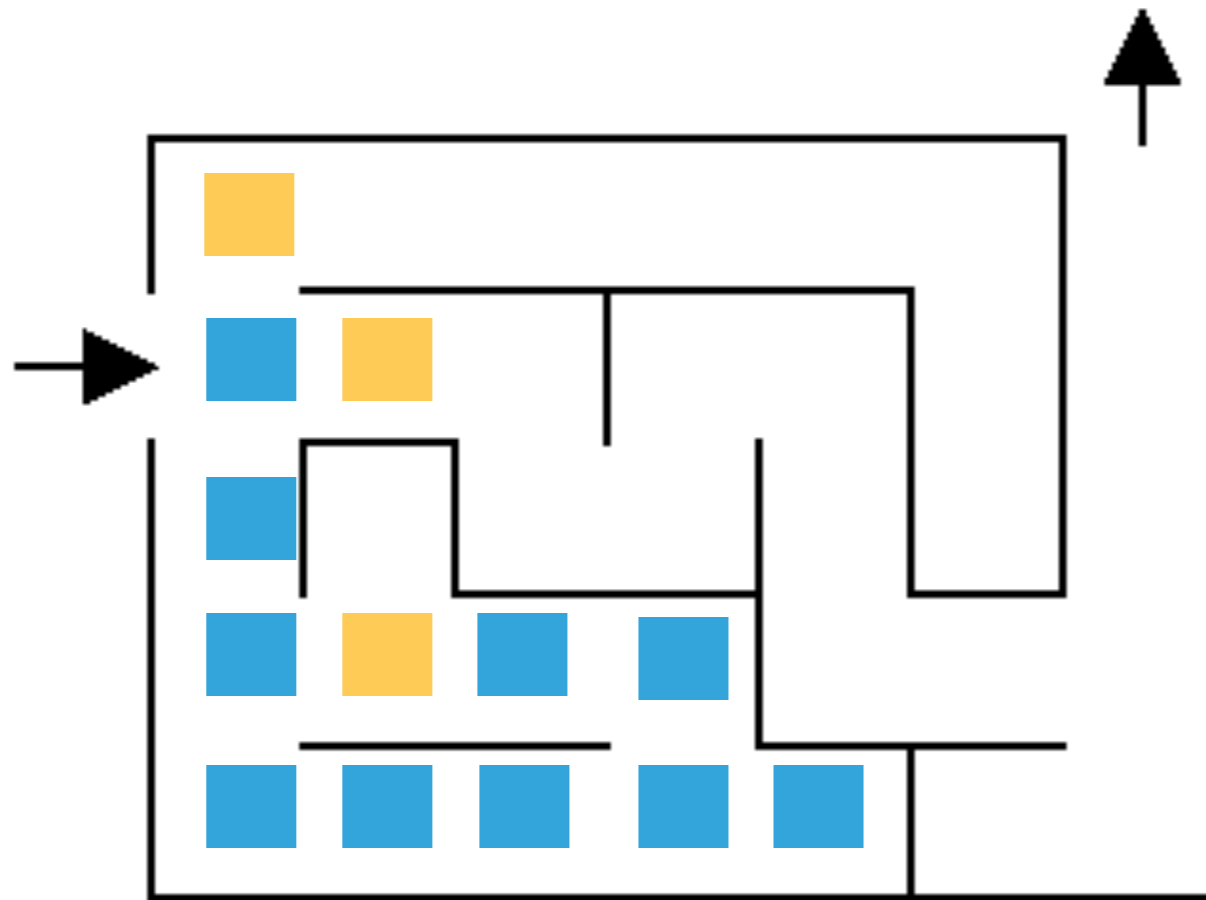
Keep exploring

One approach



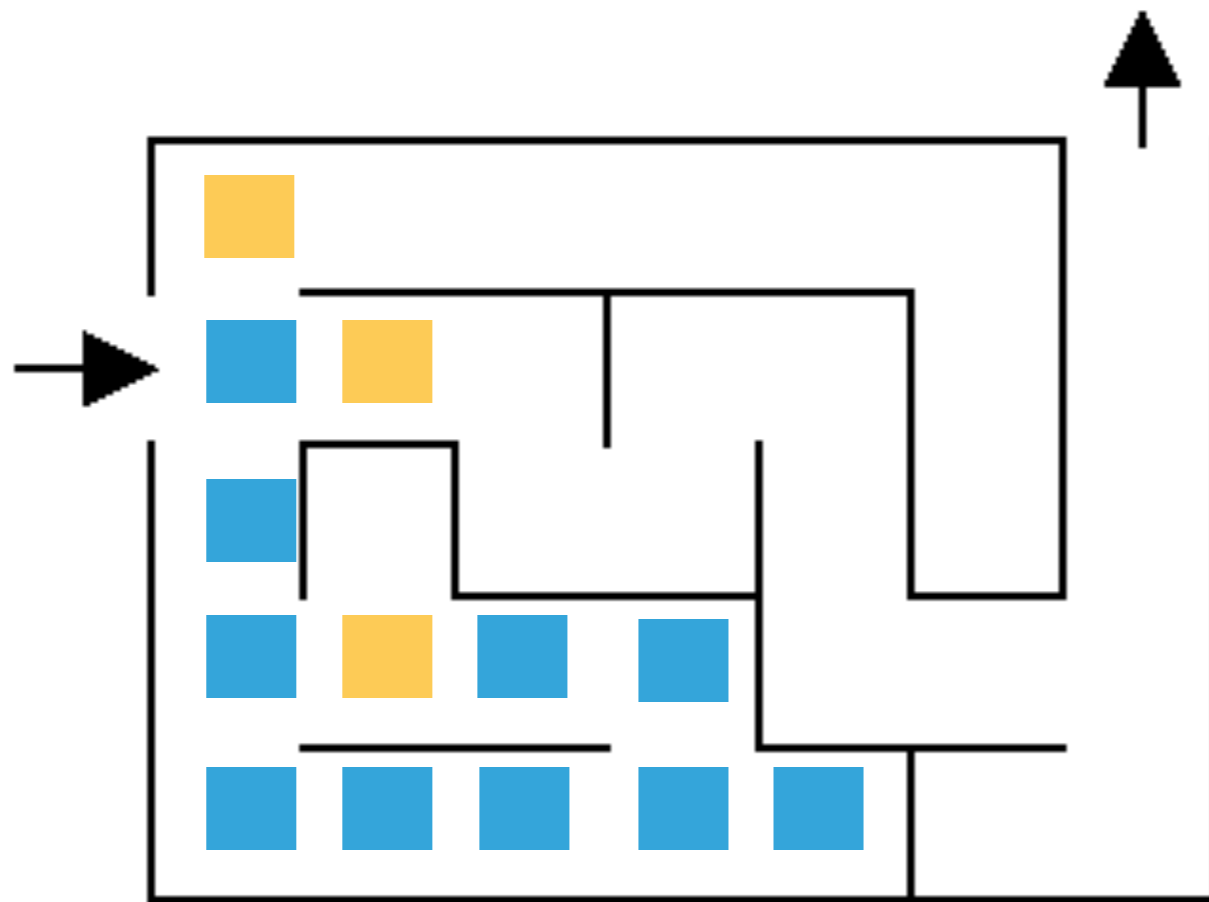
Keep exploring

One approach



Are we stuck?

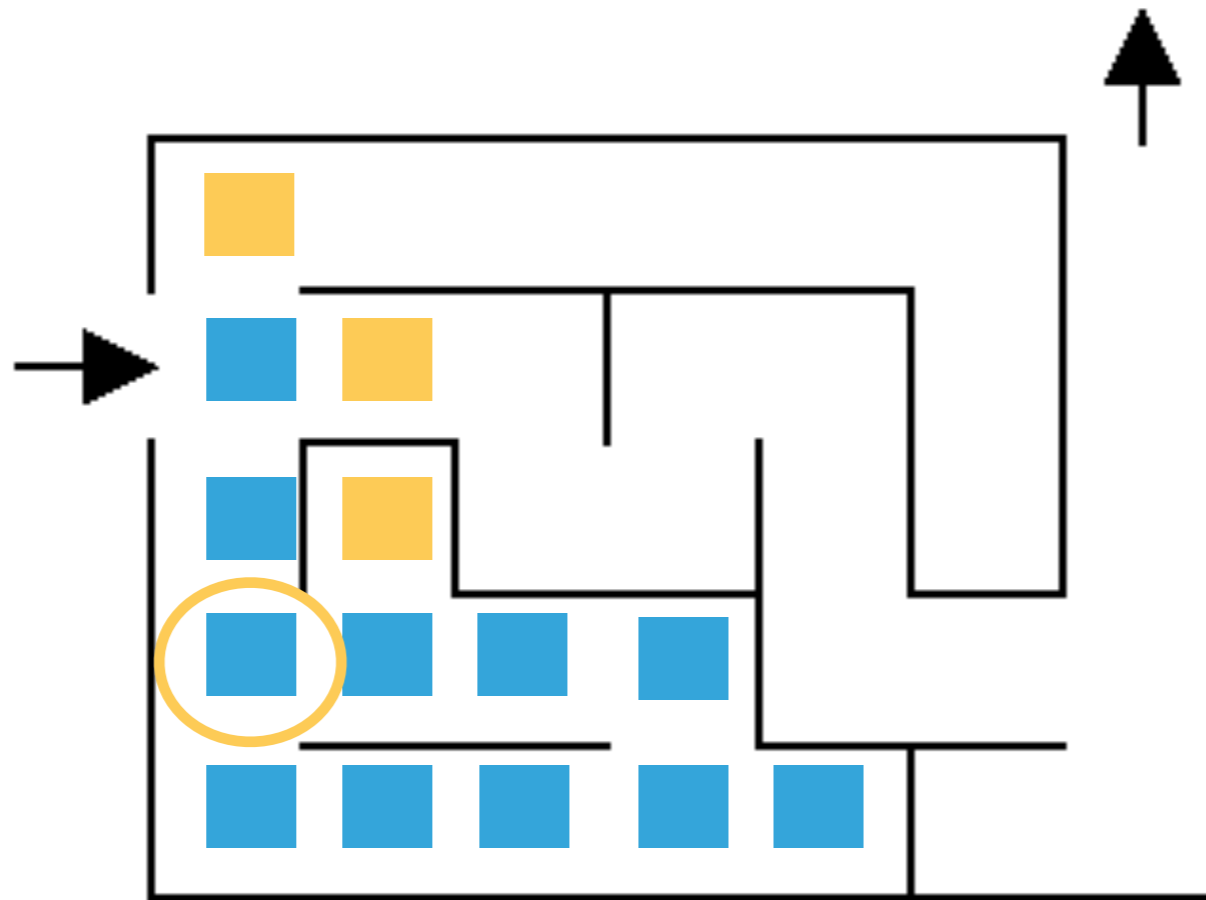
One approach



Are we stuck?

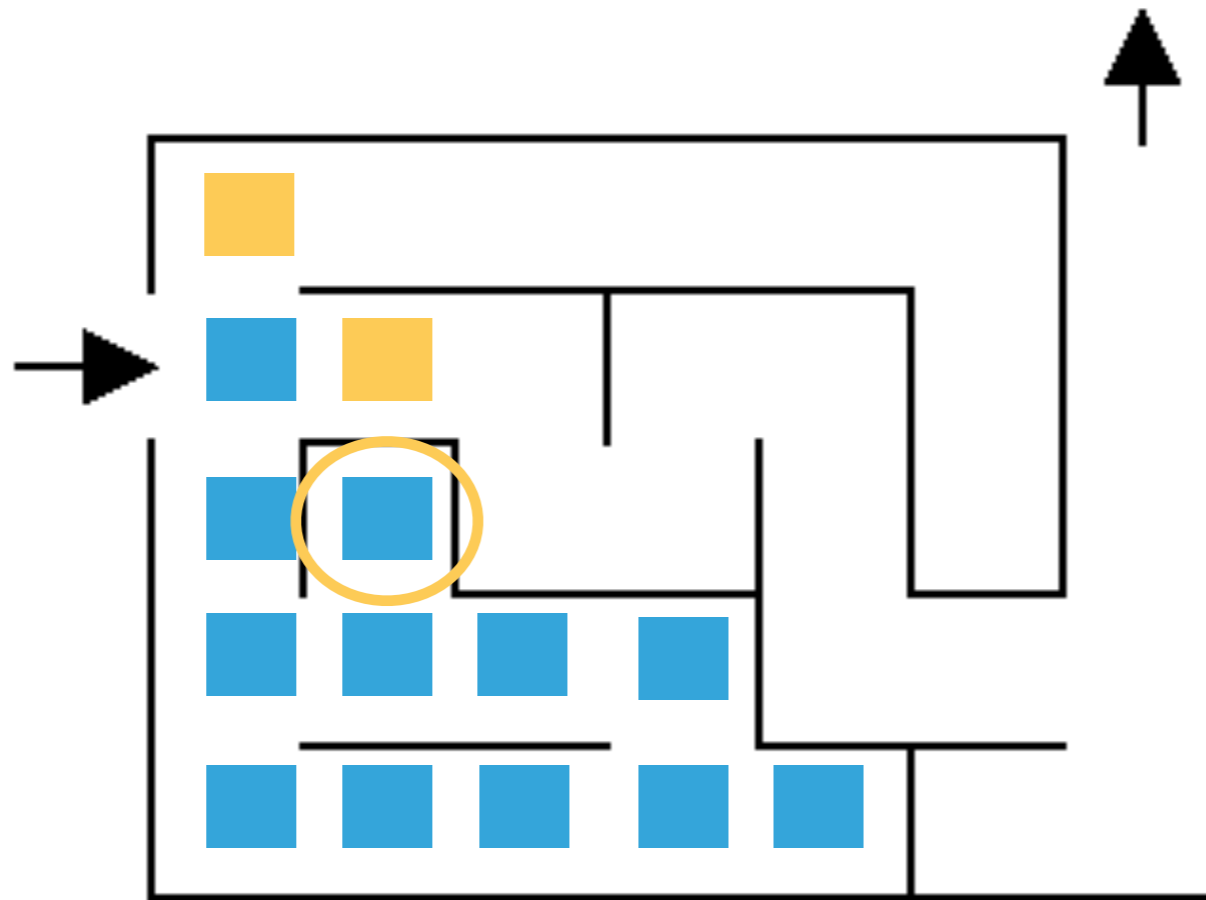
No. Yellow positions are just possible options we haven't explored

One approach



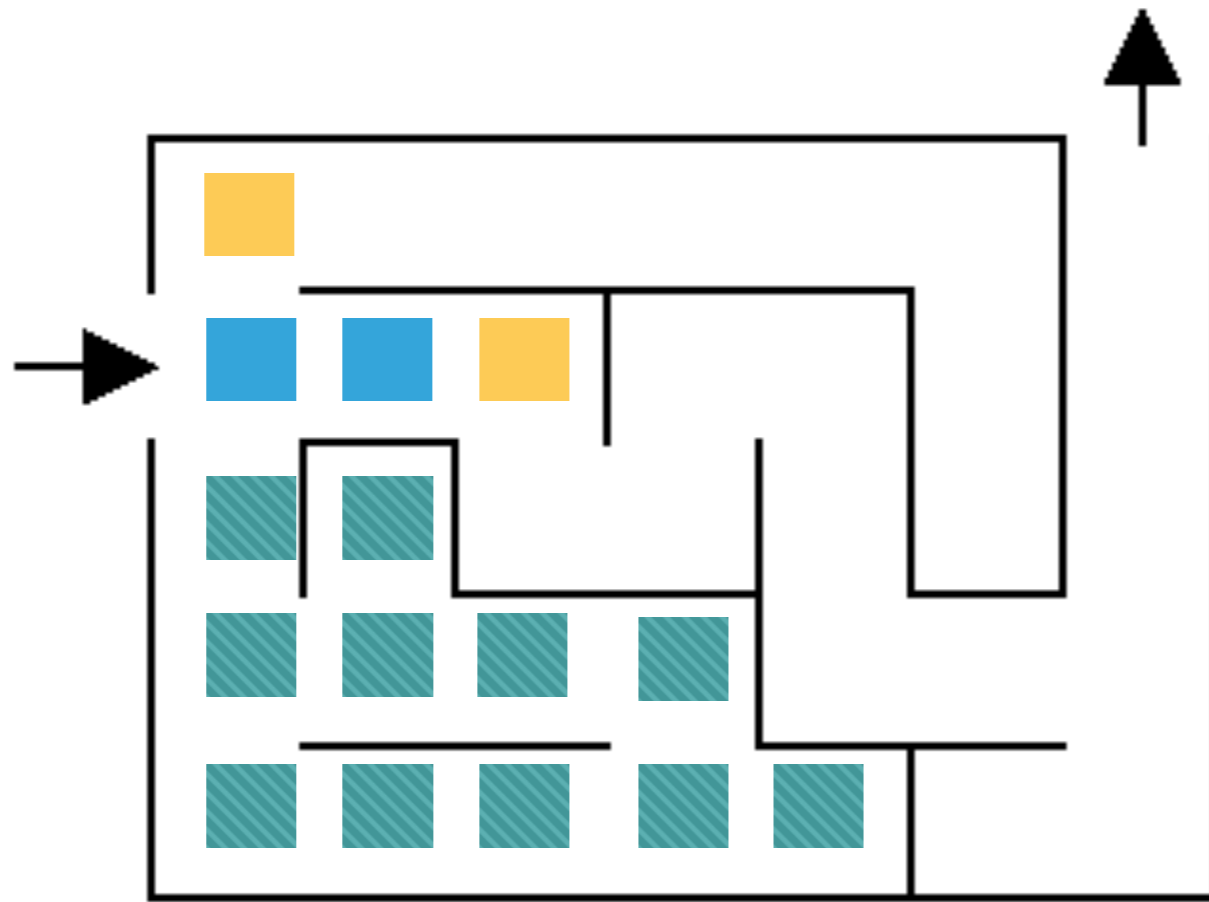
How do we know not to go here?

One approach



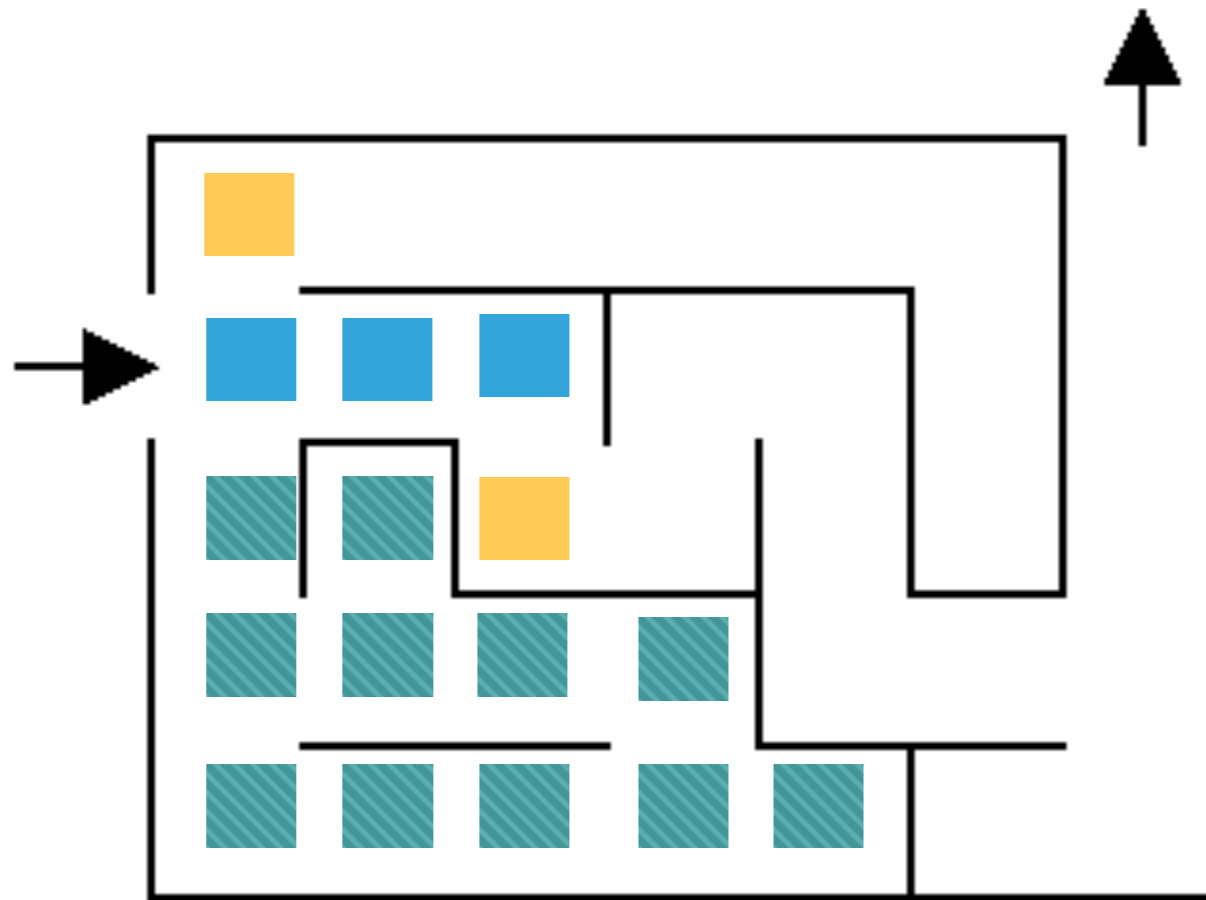
What now?

One approach



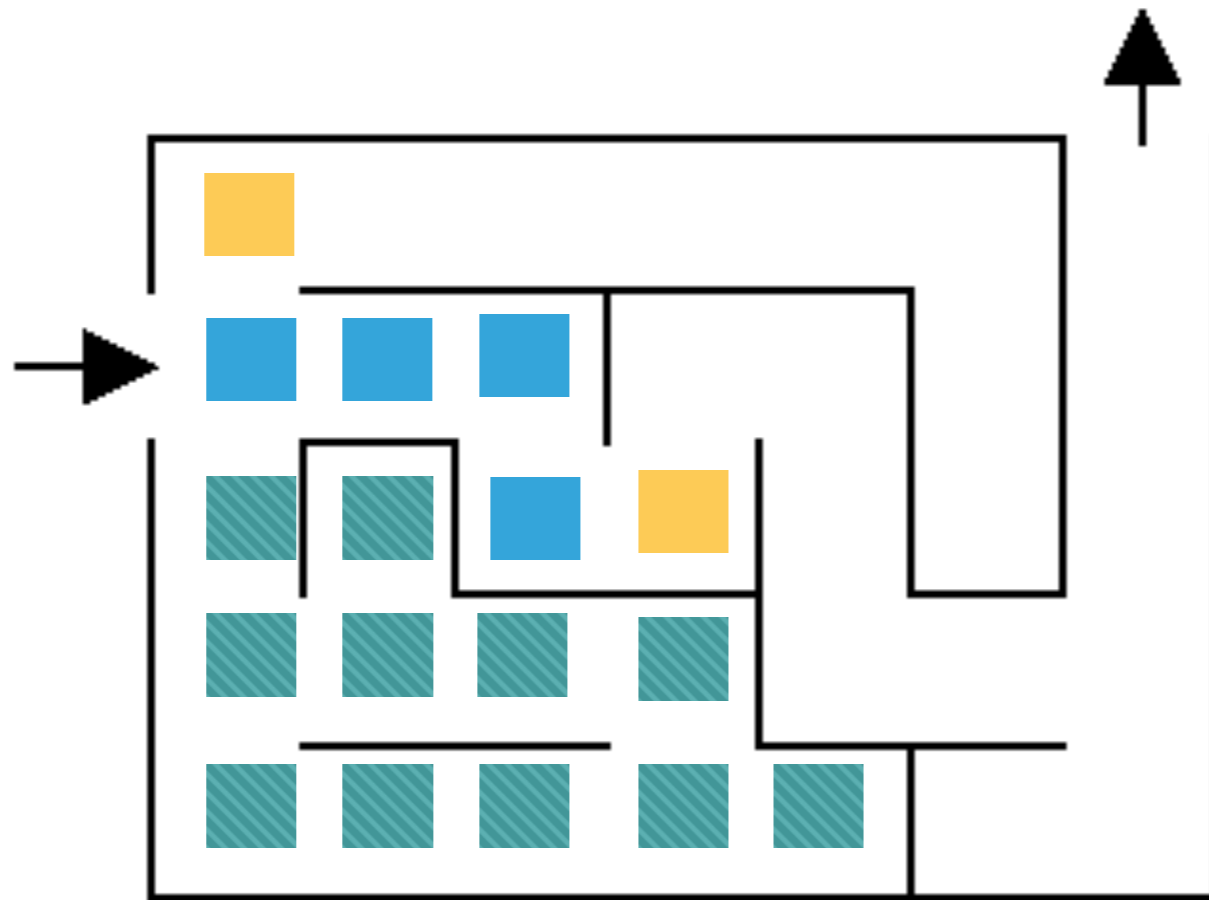
Keep exploring

One approach



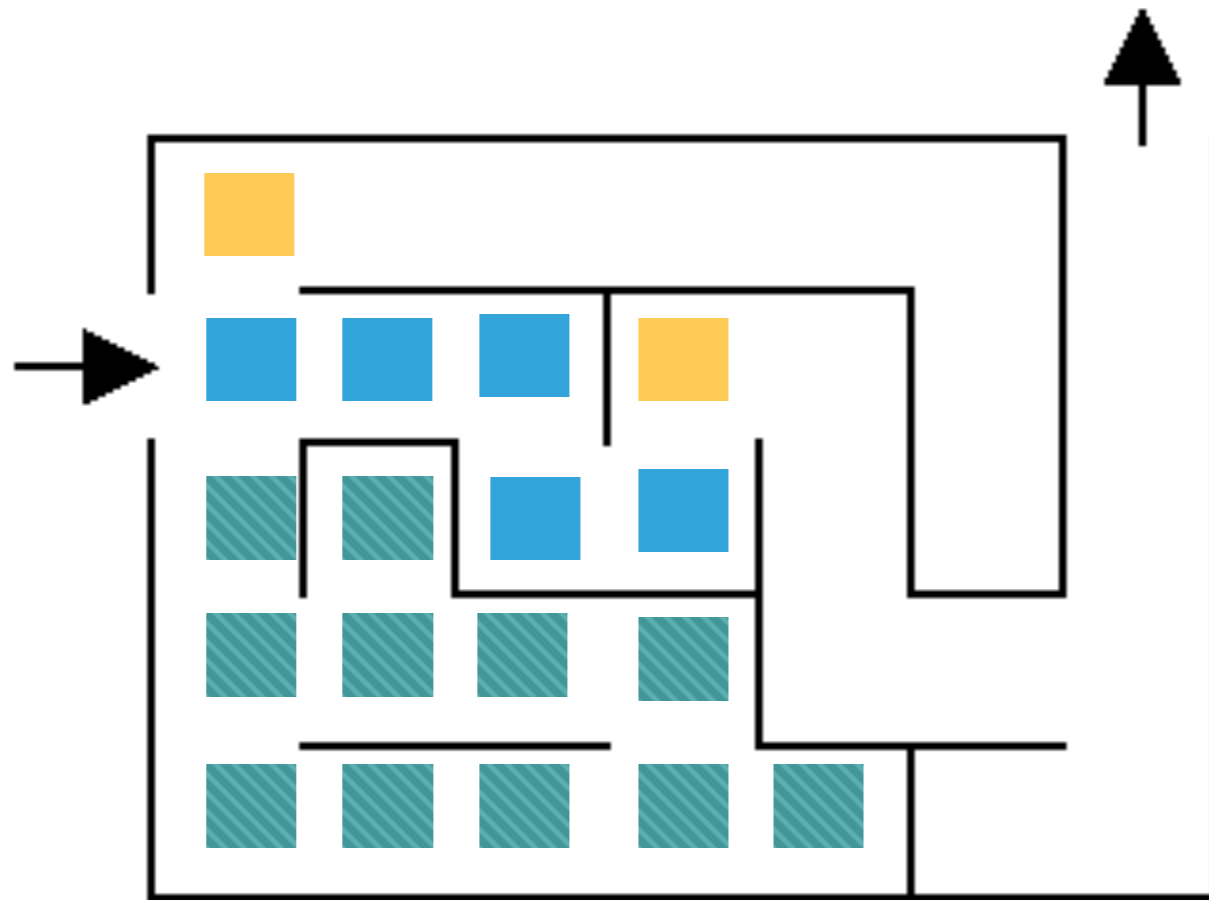
Keep exploring

One approach



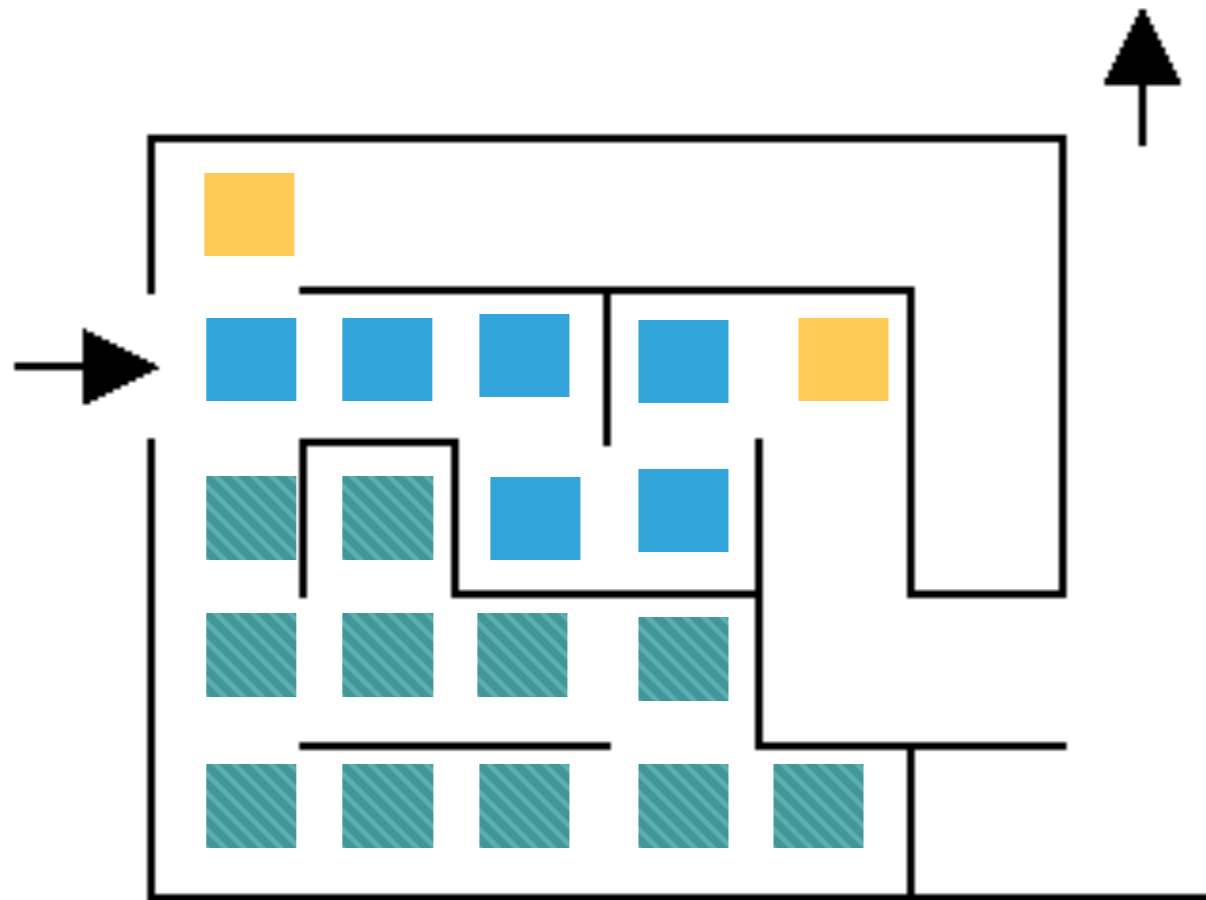
Keep exploring

One approach



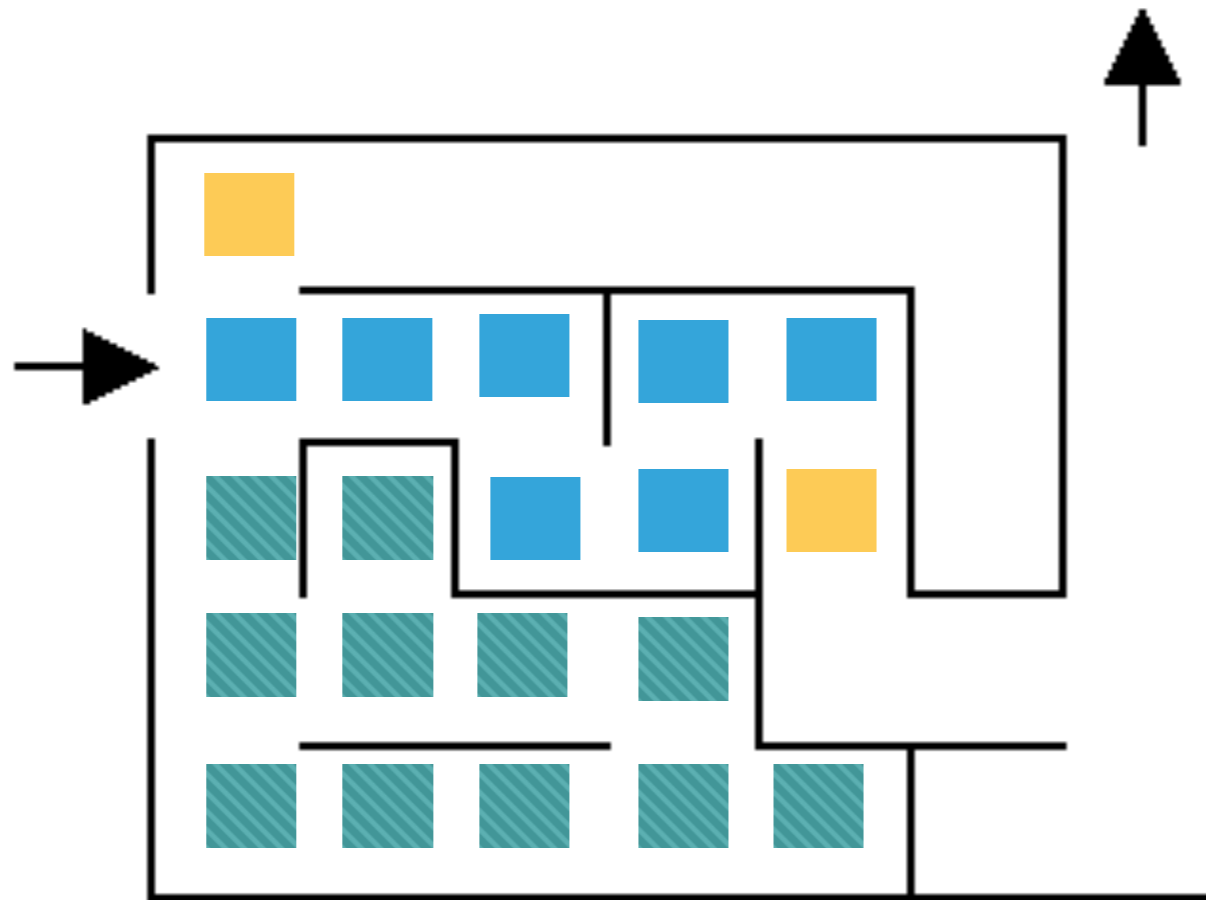
Keep exploring

One approach



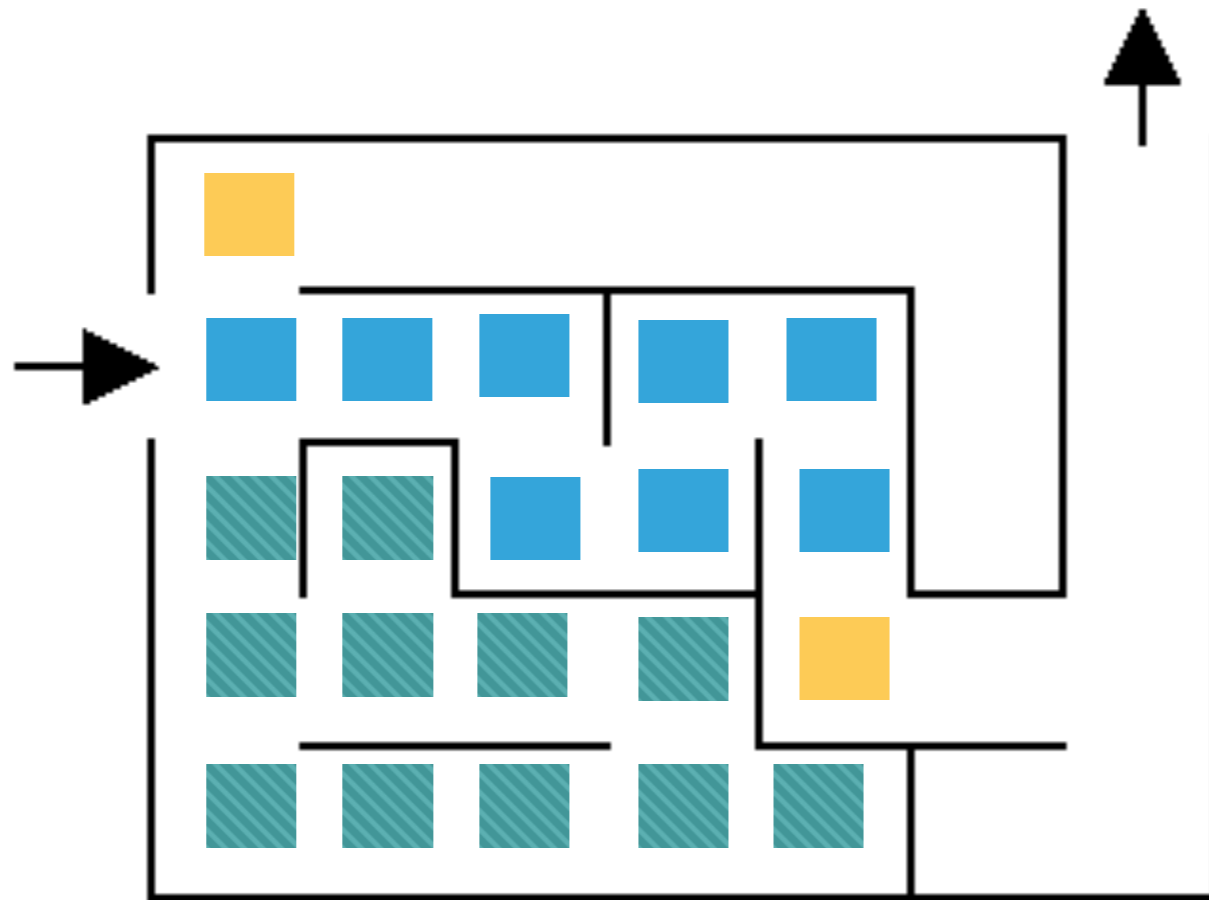
Keep exploring

One approach



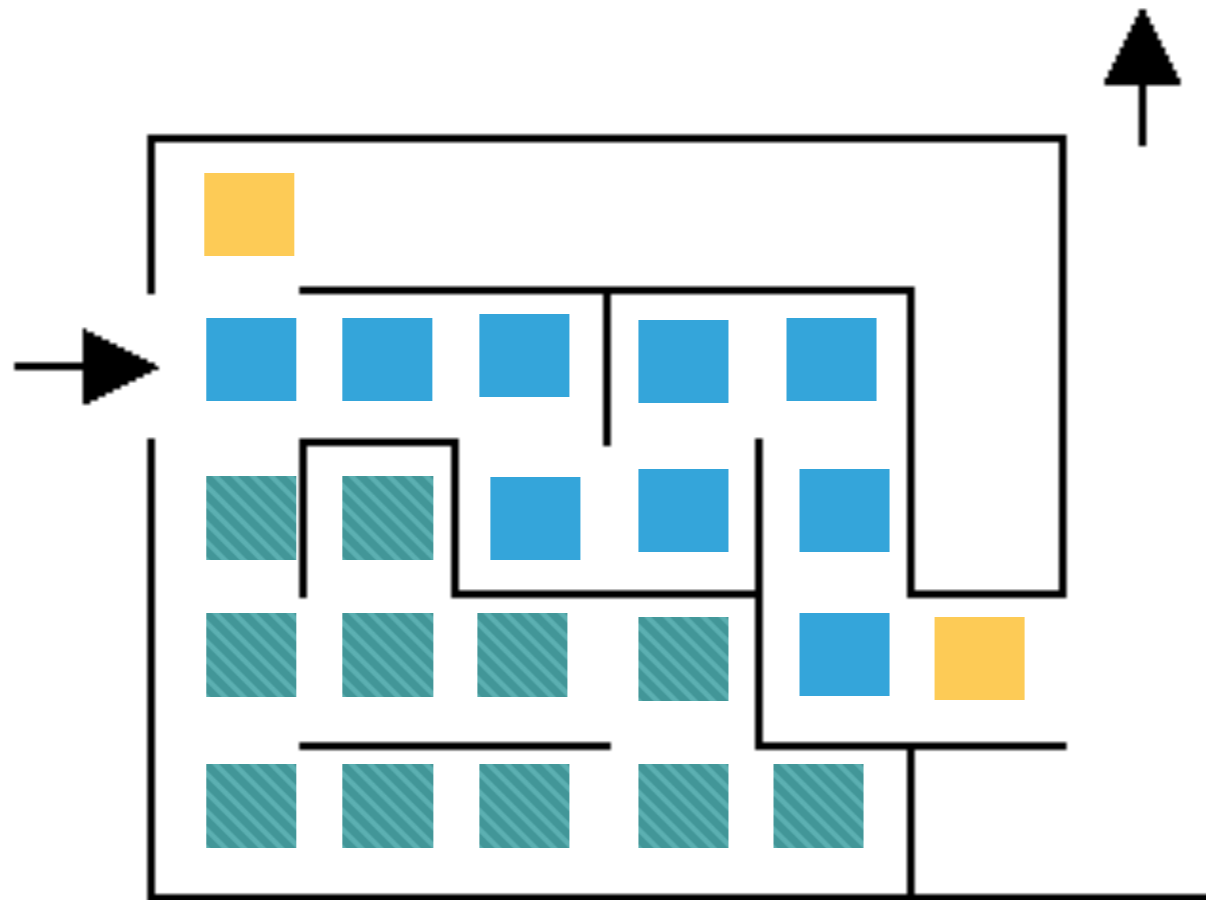
Keep exploring

One approach



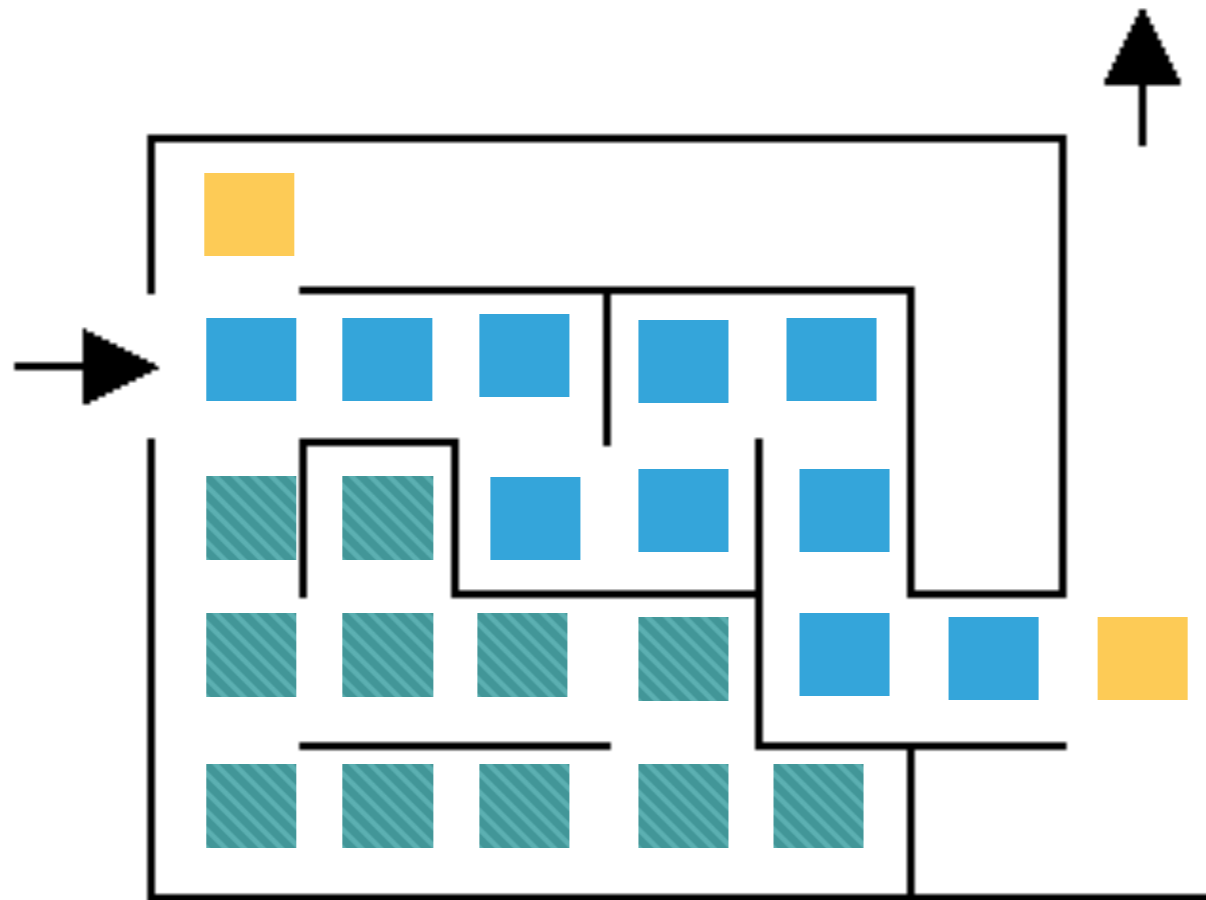
Keep exploring

One approach



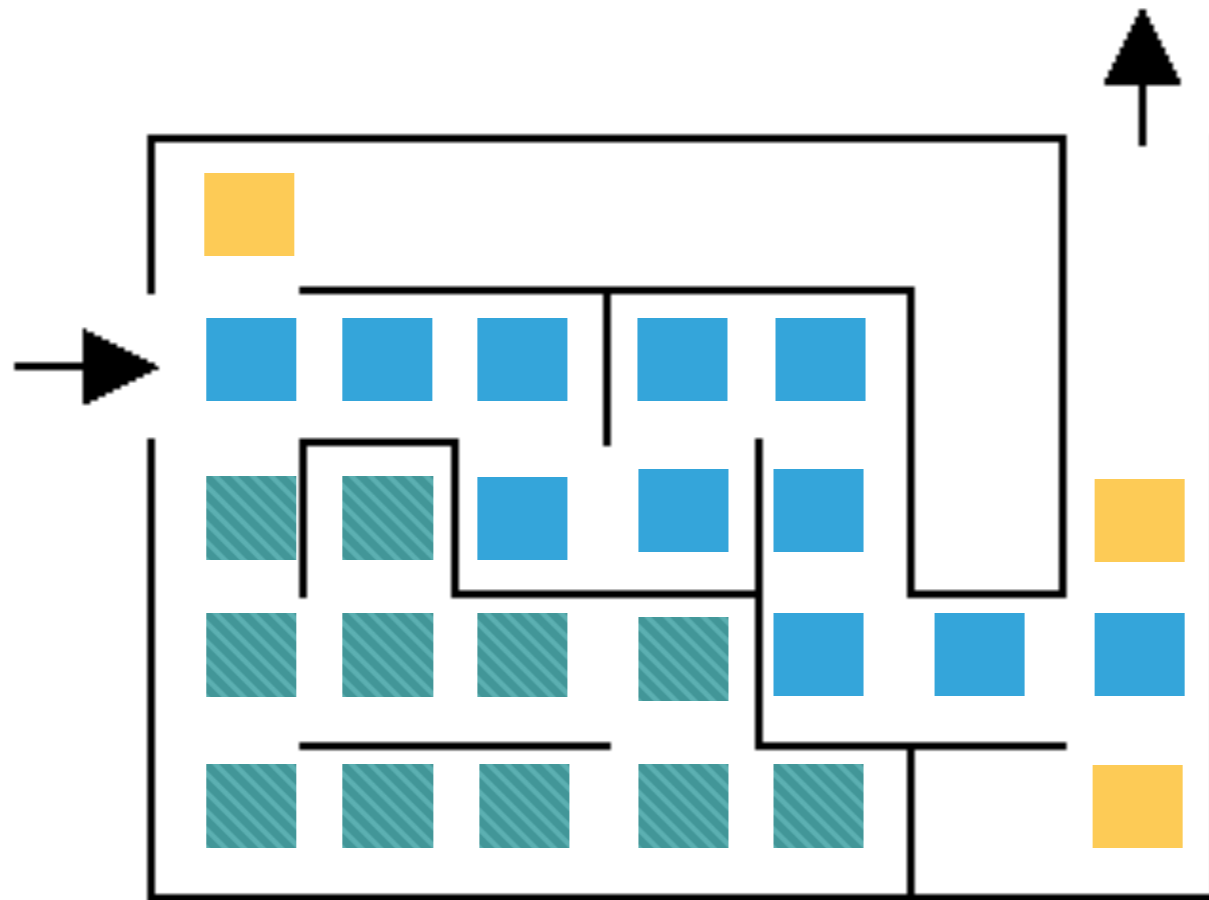
Keep exploring

One approach



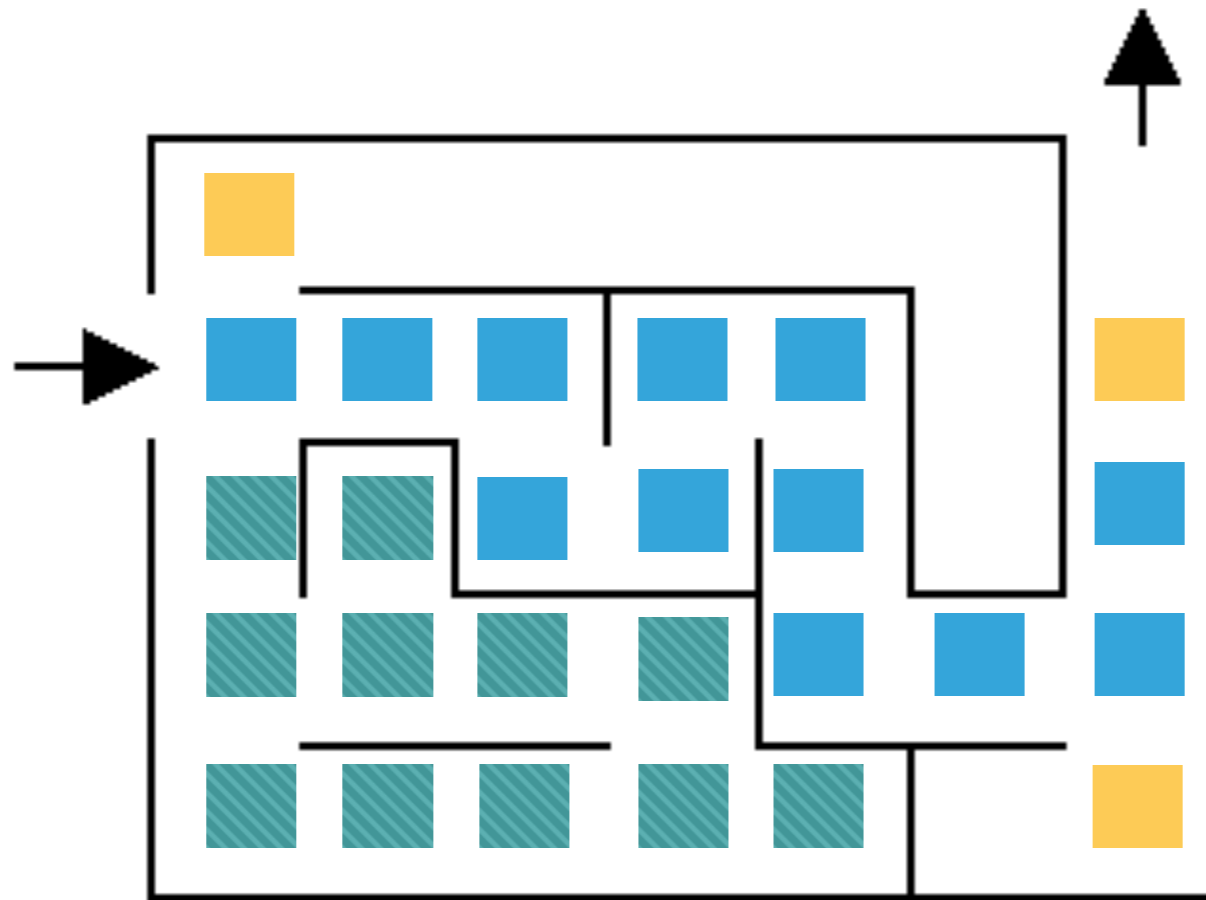
Keep exploring

One approach



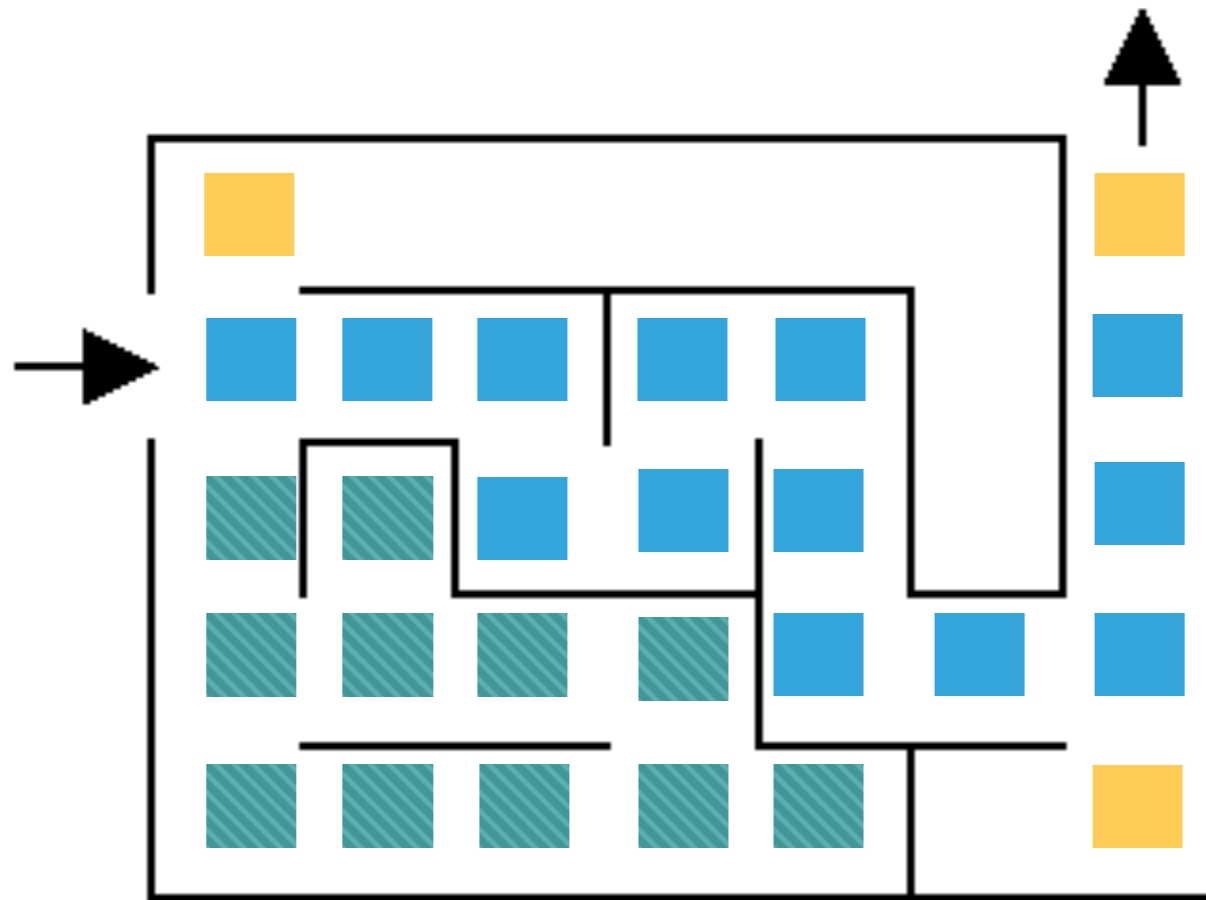
What now?

One approach



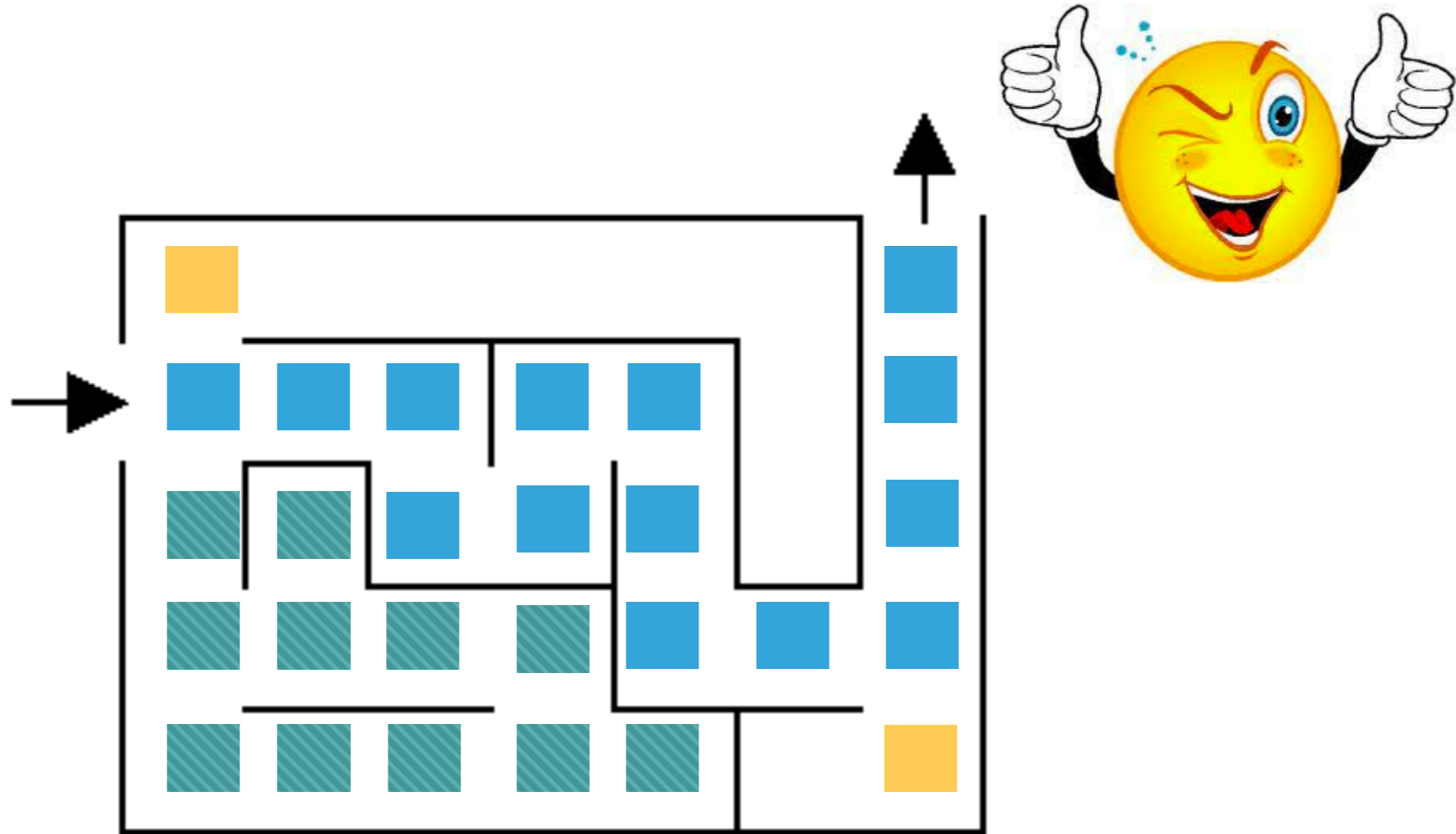
Keep exploring

One approach

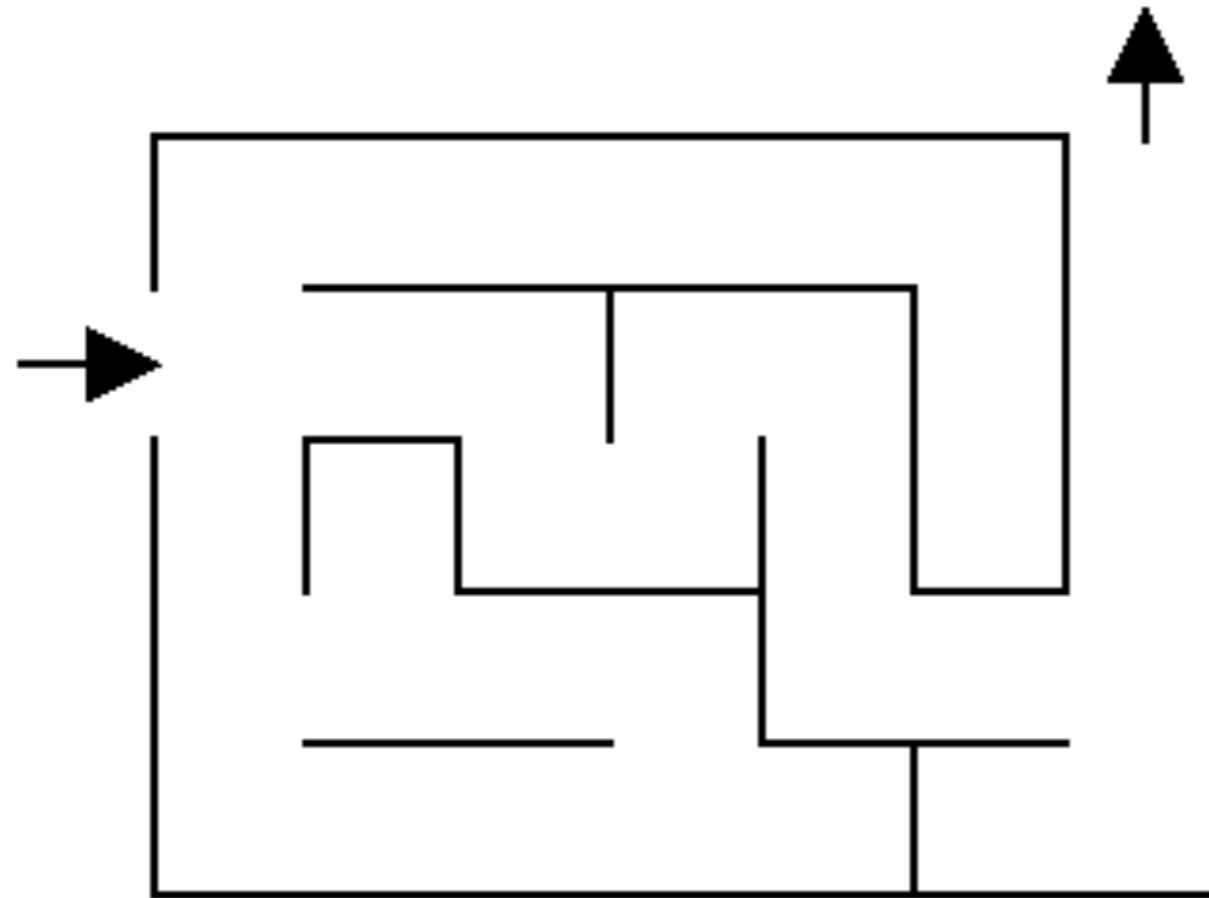


Keep exploring

One approach



Search problems

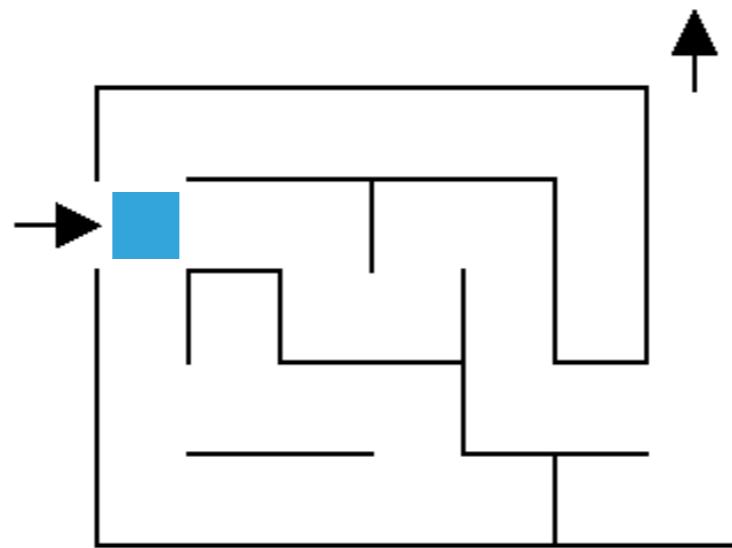


What information do we need to figure out a solution?

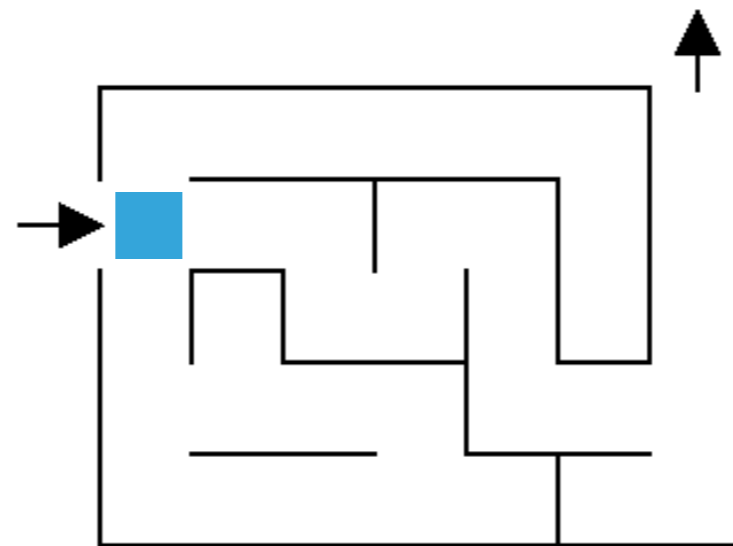
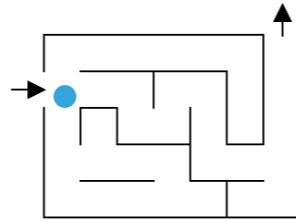
Search problems

- ▶ Where to start.
- ▶ Where to finish (goal).
- ▶ What the “world” (in this case a maze) looks like.
 - ▶ We’ll define the world as a collection of **discrete** states.
 - ▶ States are connected if we can get from one state to another by taking a particular action.
 - ▶ The set of all possible states is called the **state space**.

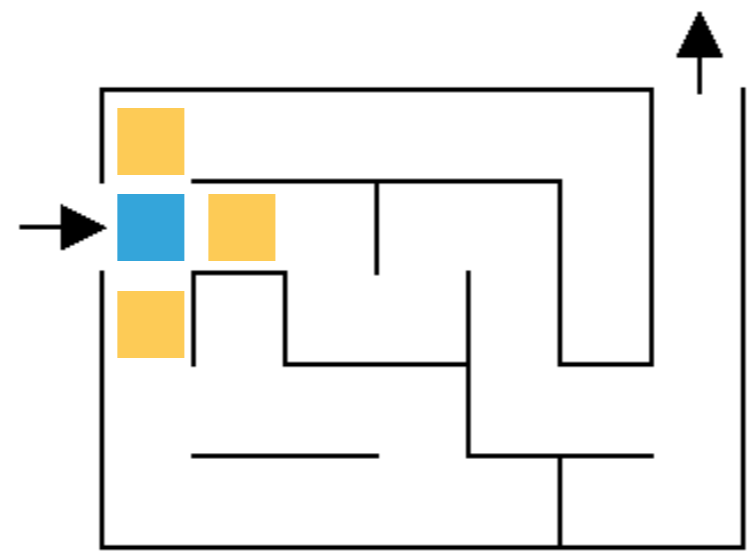
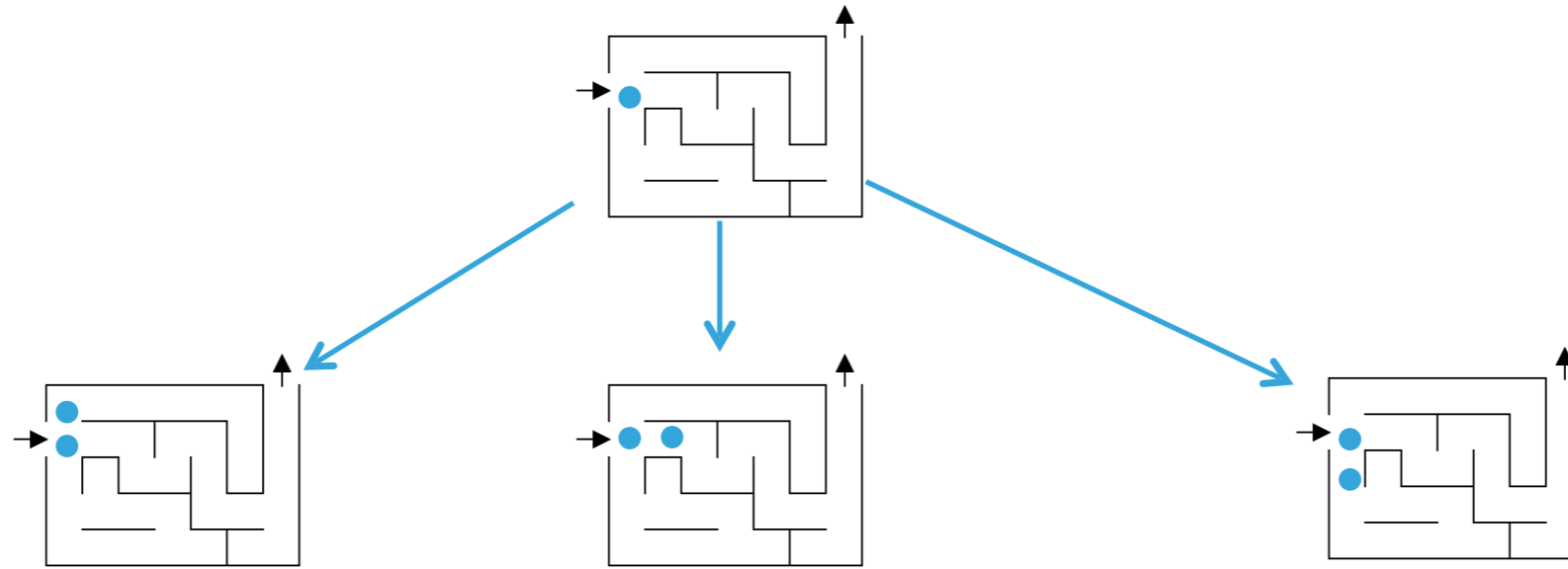
State space example



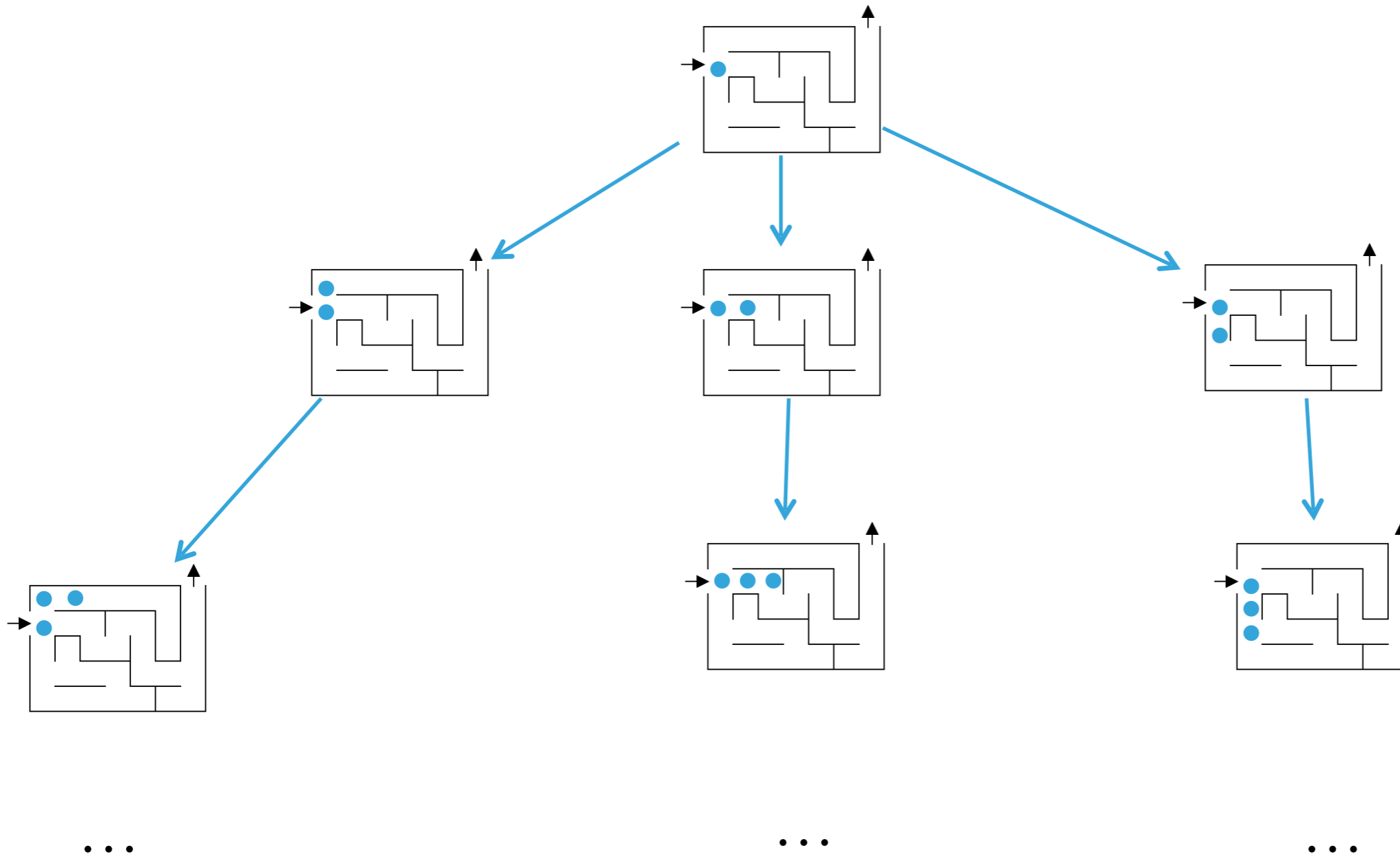
State space example



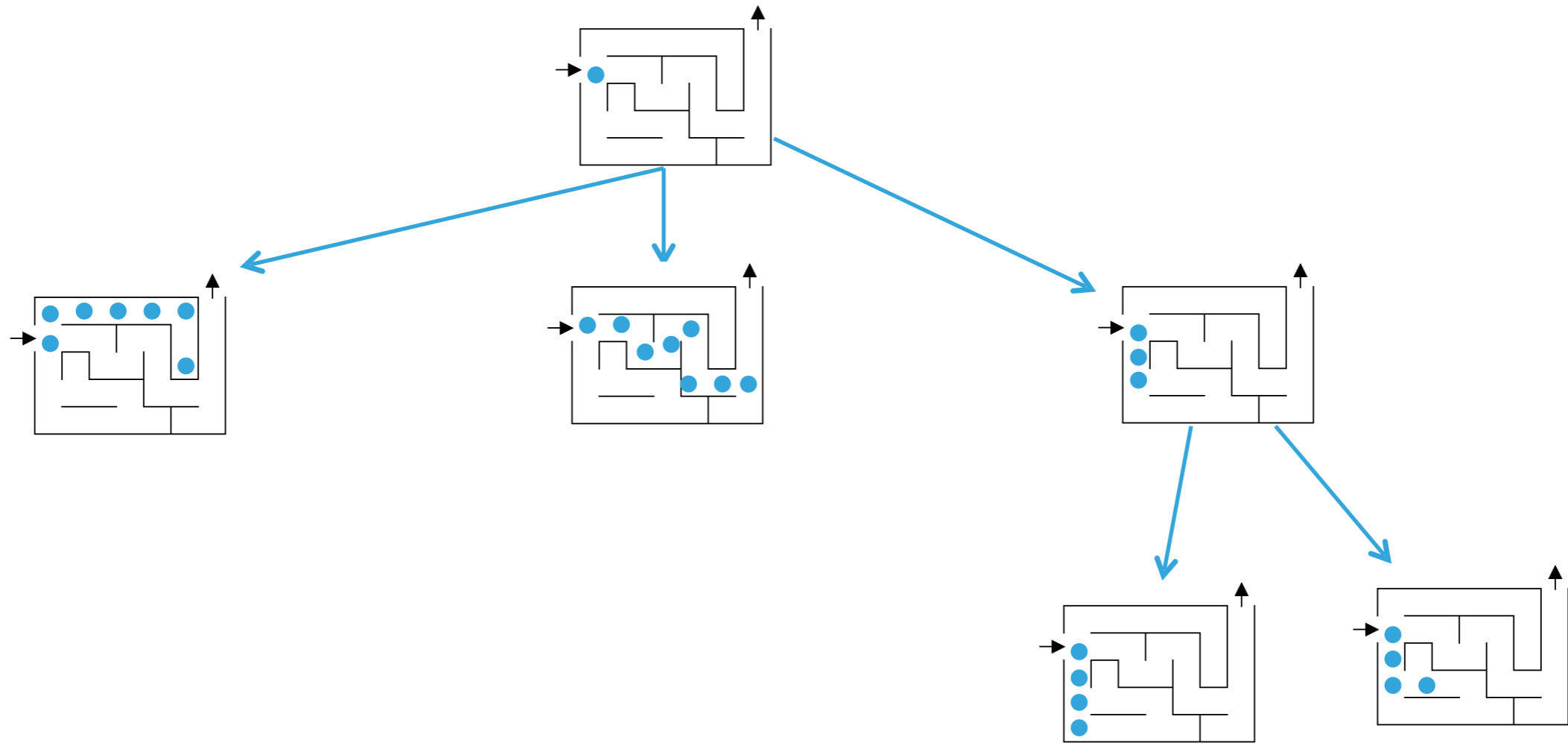
State space example



State space example

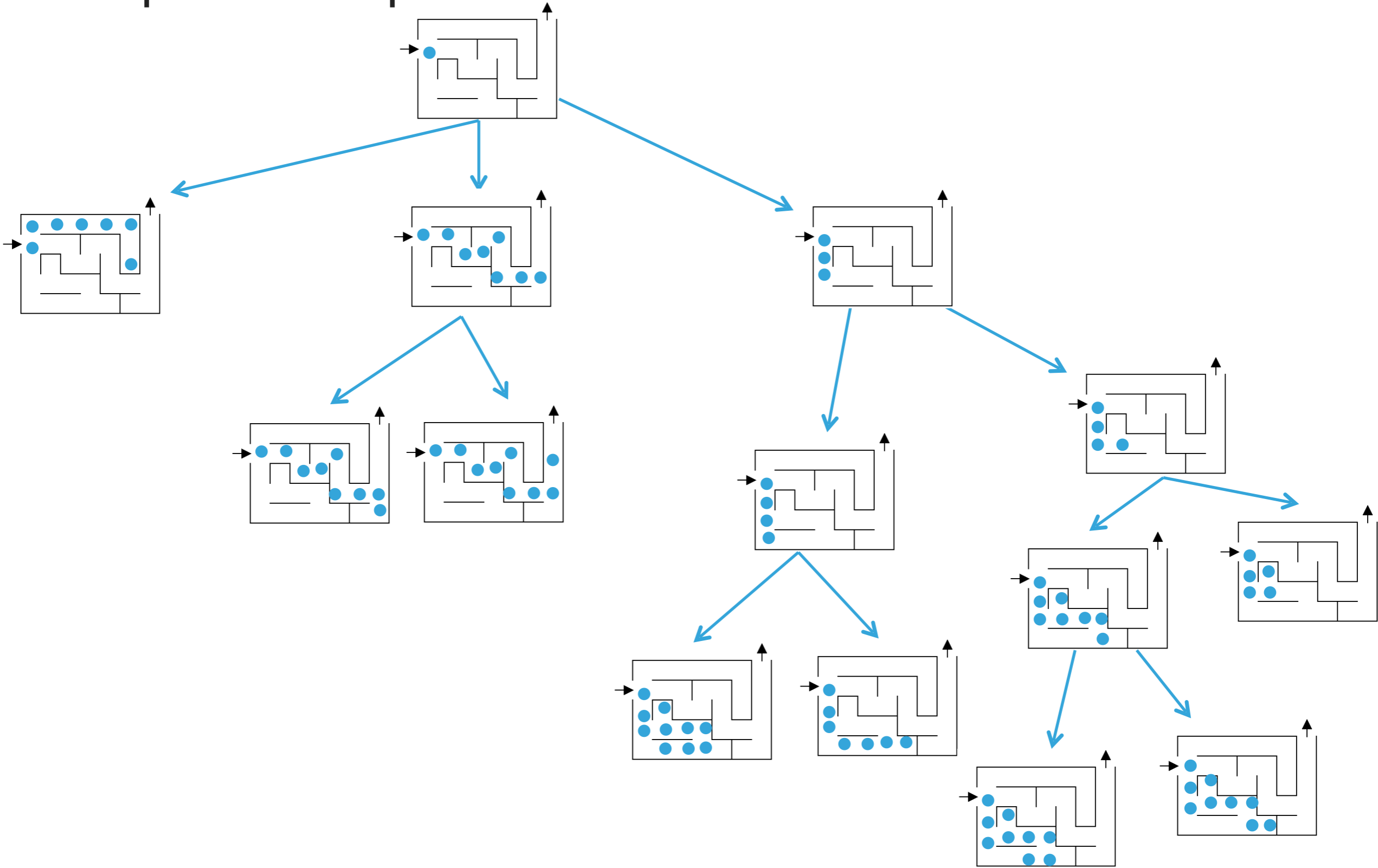


State space example

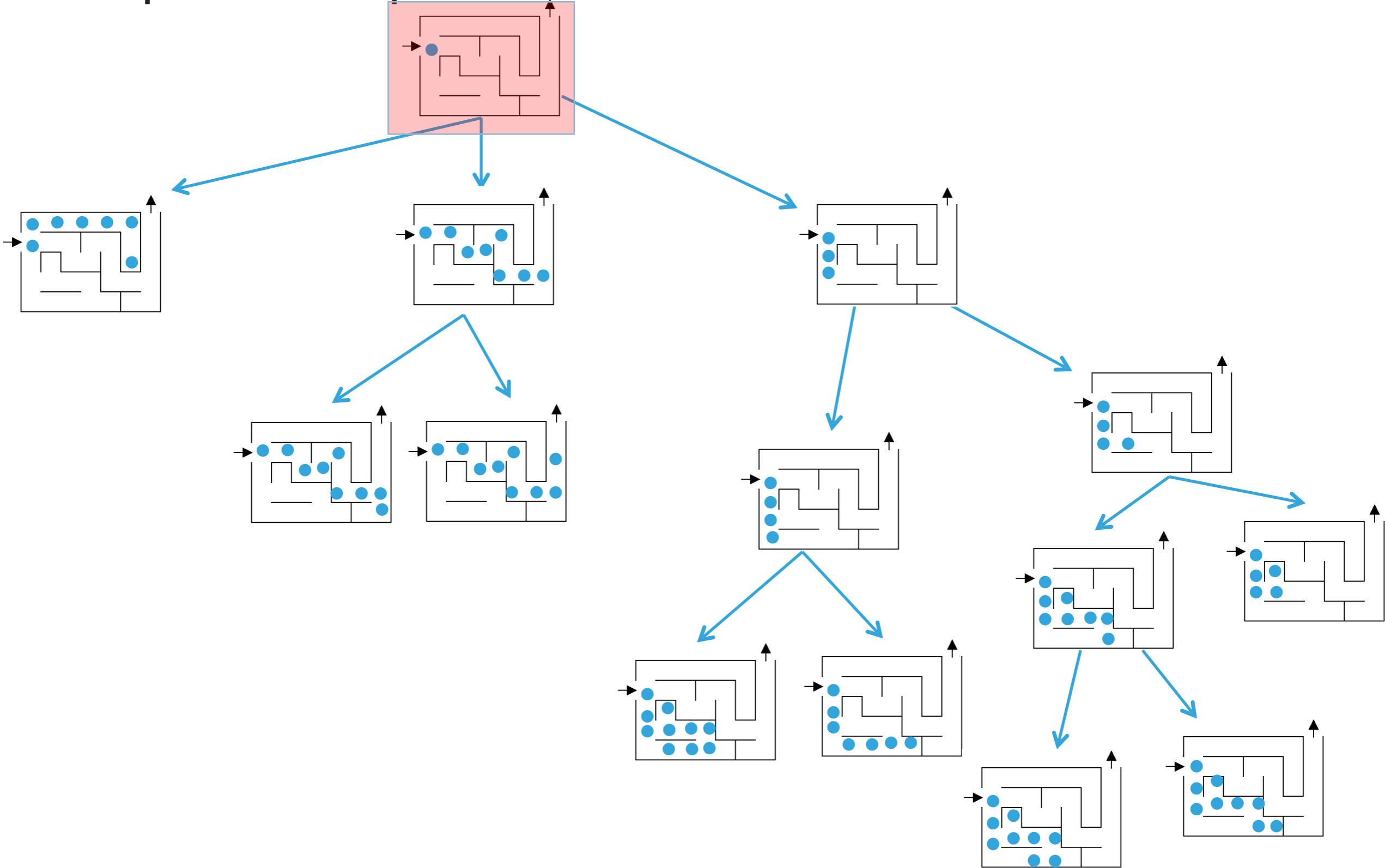


How many more states are there?

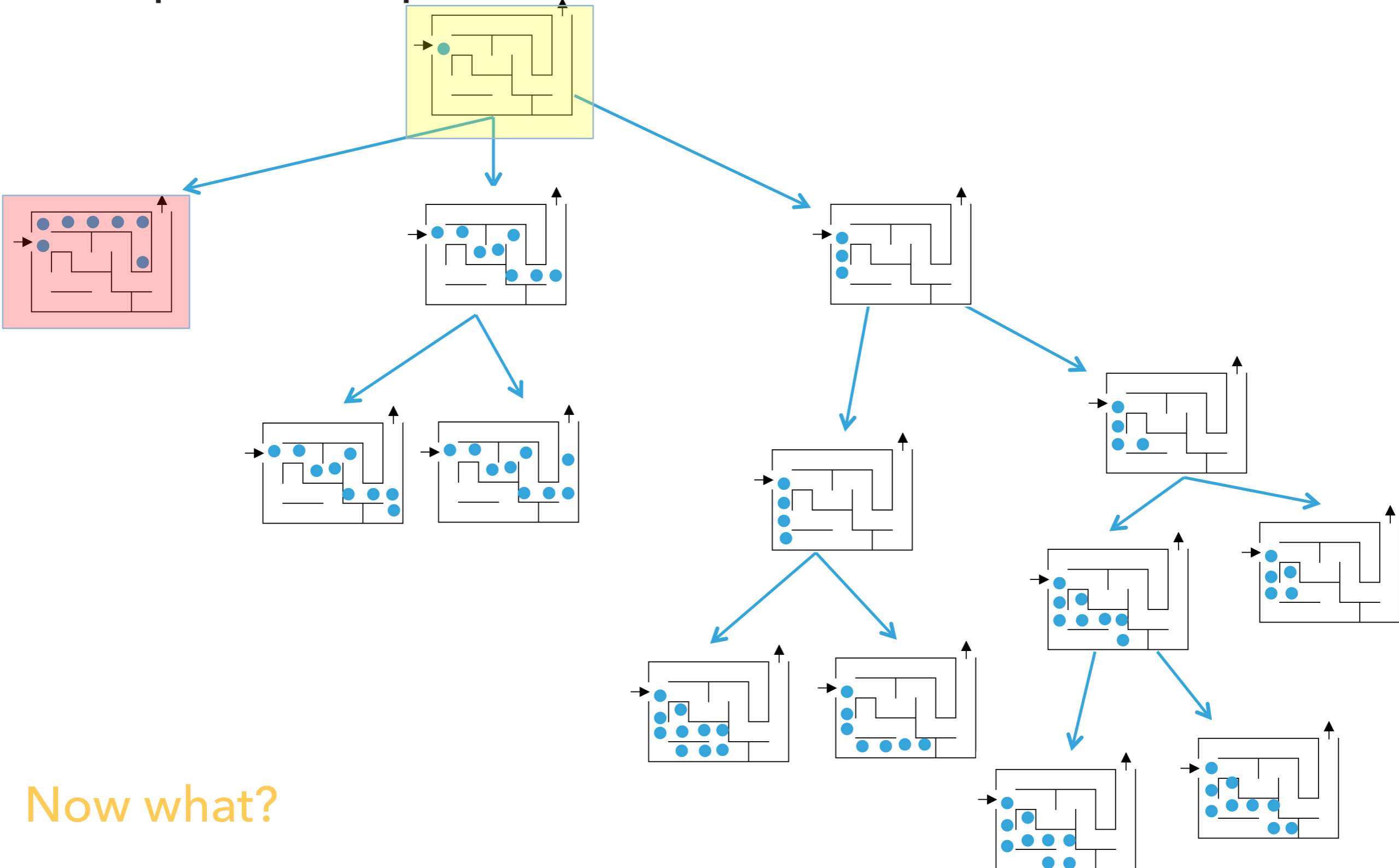
State space example



State space example

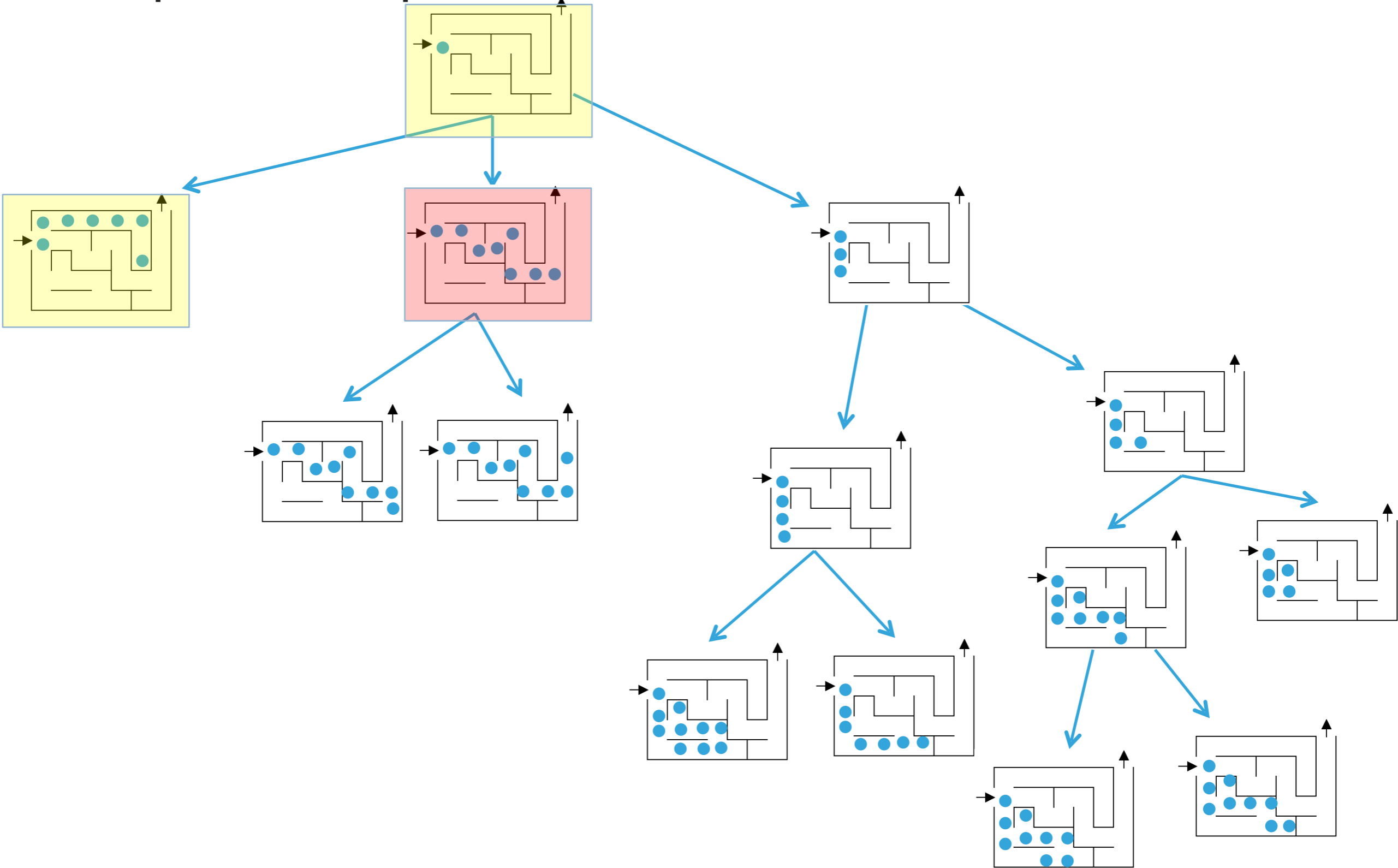


State space example

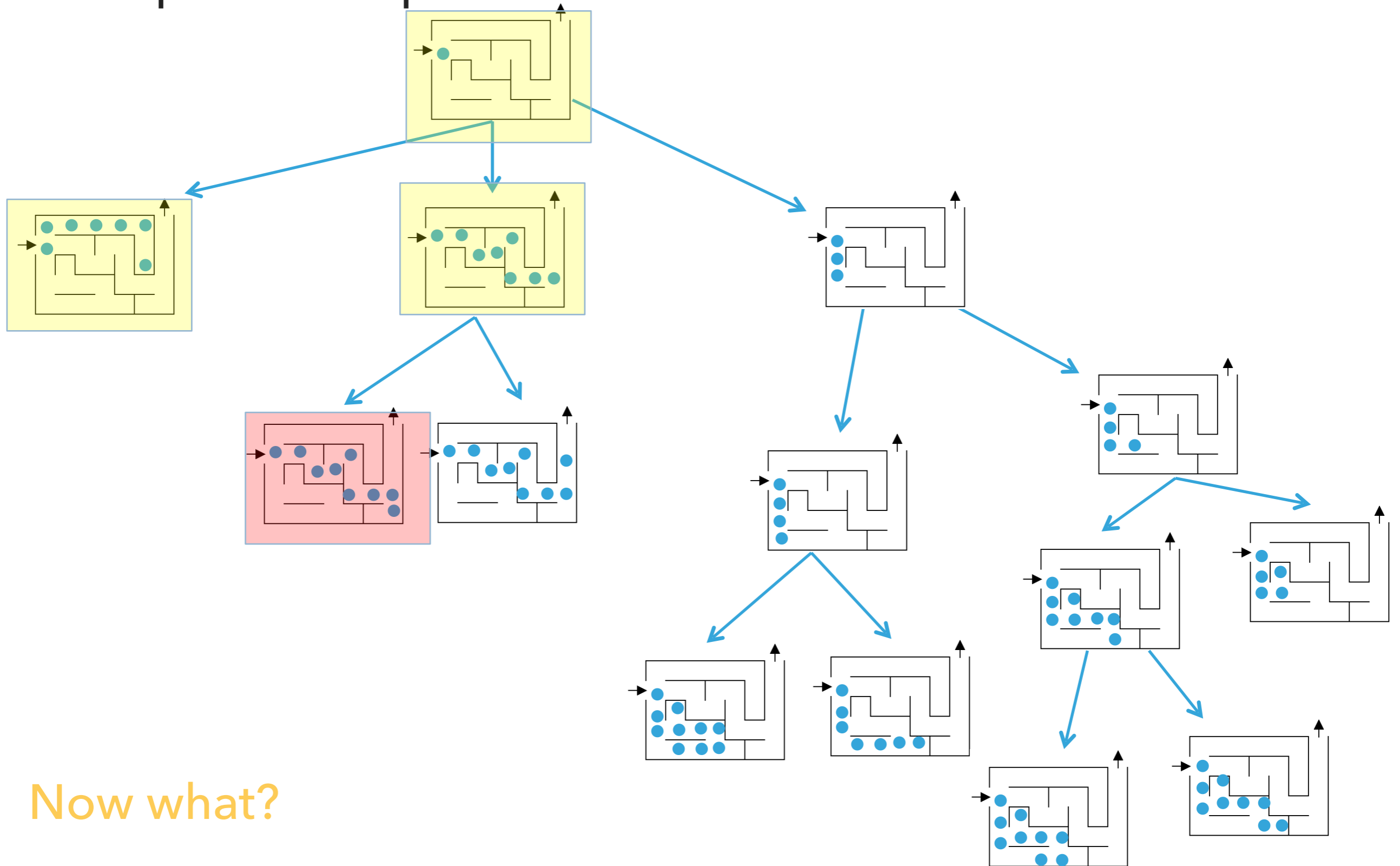


Now what?

State space example

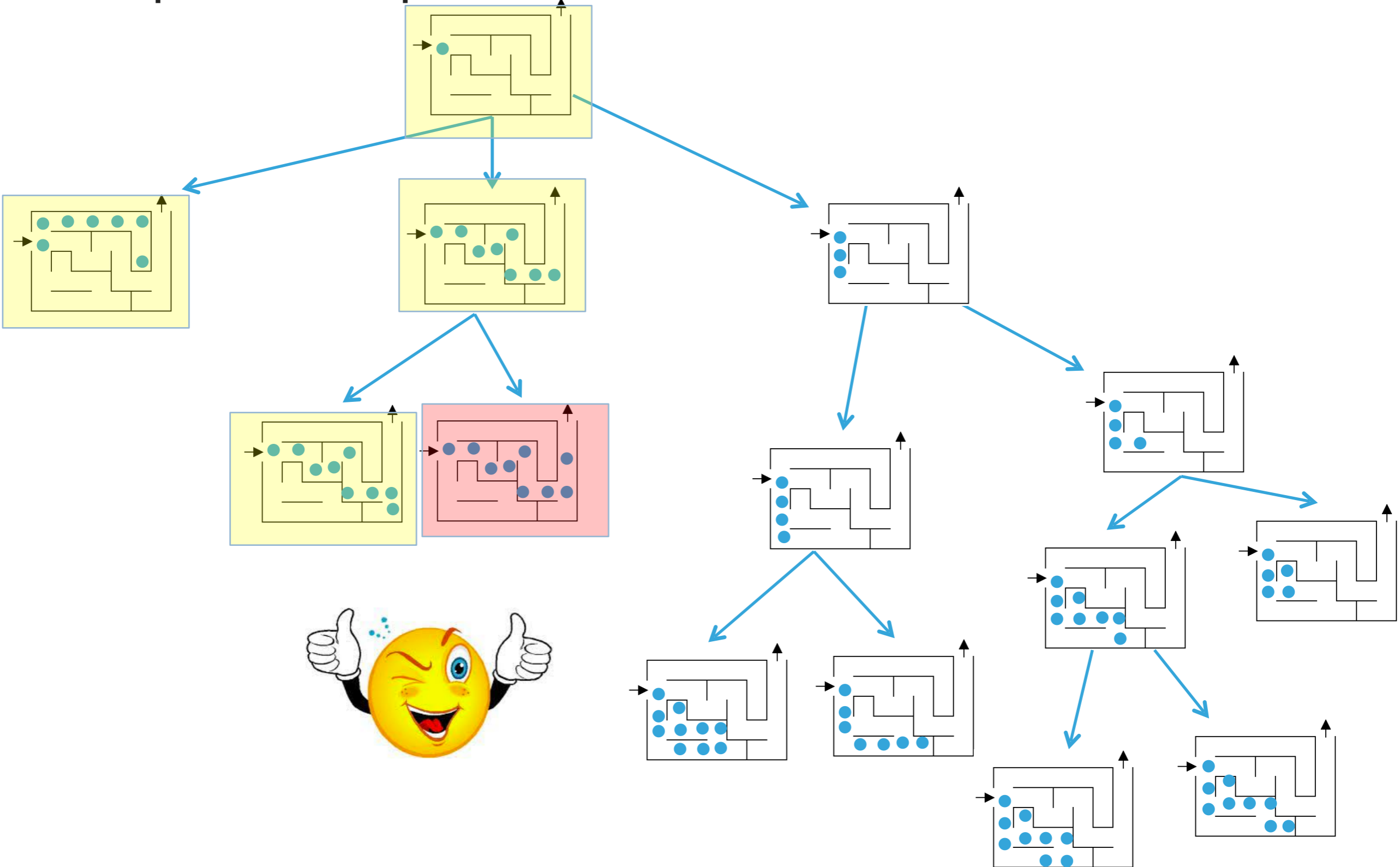


State space example

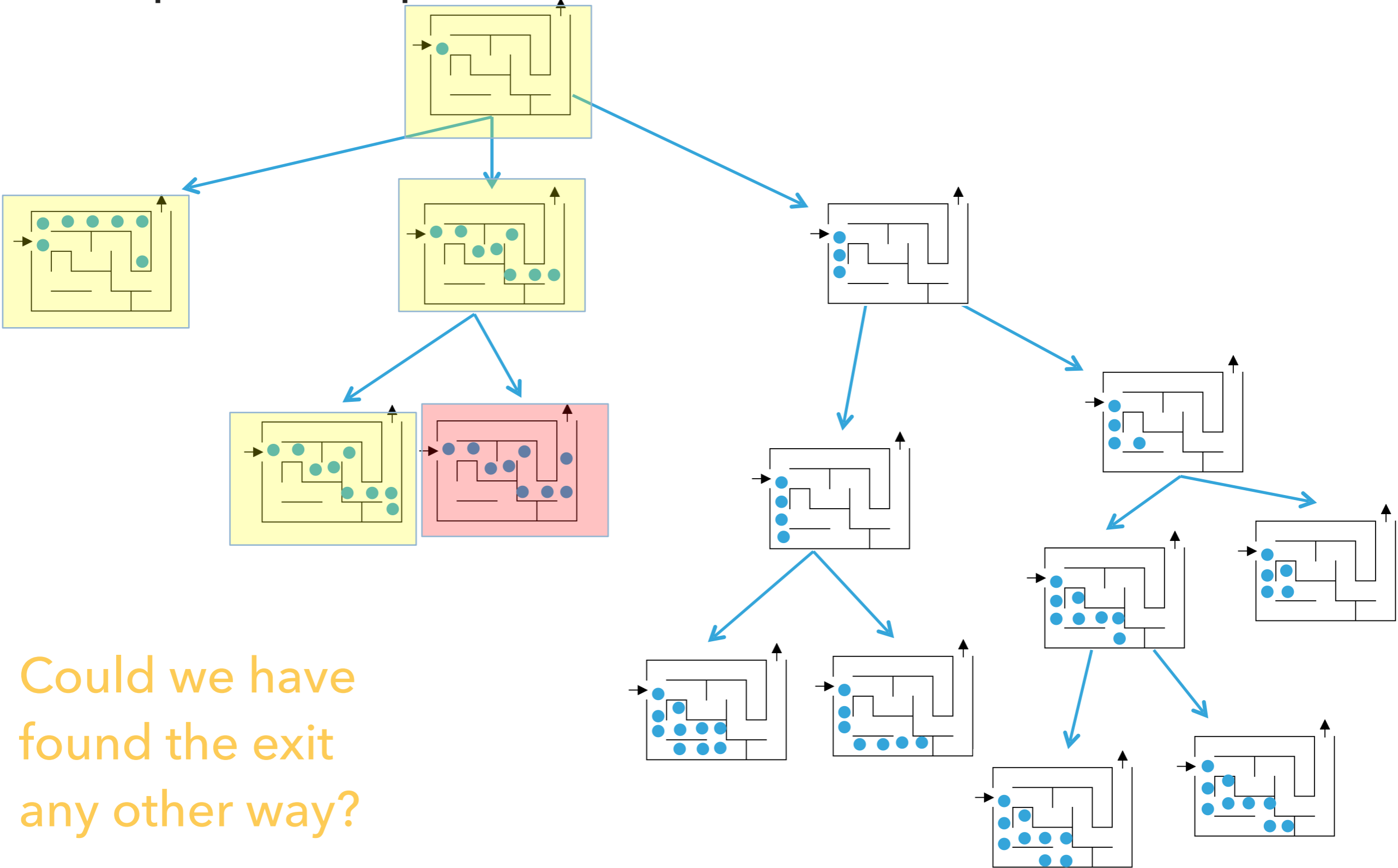


Now what?

State space example



State space example



Could we have found the exit any other way?

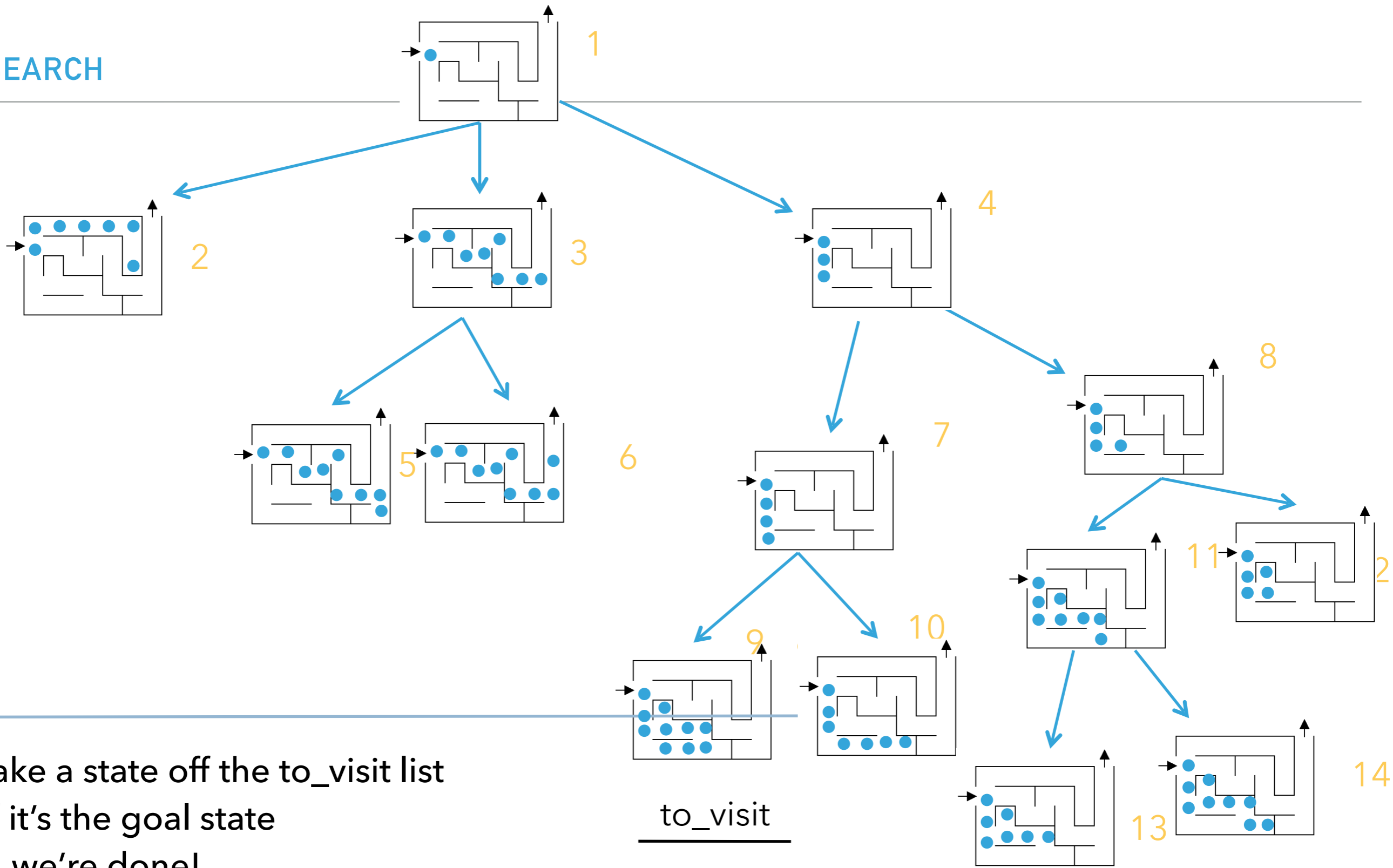
Search algorithm

Keep track of a list of states that we *could* visit; we'll call it **to_visit**.

General idea:

- ▶ take a state off the to_visit list
- ▶ if it's the goal state
 - ▶ we're done!
- ▶ if it's not the goal state
 - ▶ Add all of the next possible states to the to_visit list
- ▶ repeat

SEARCH

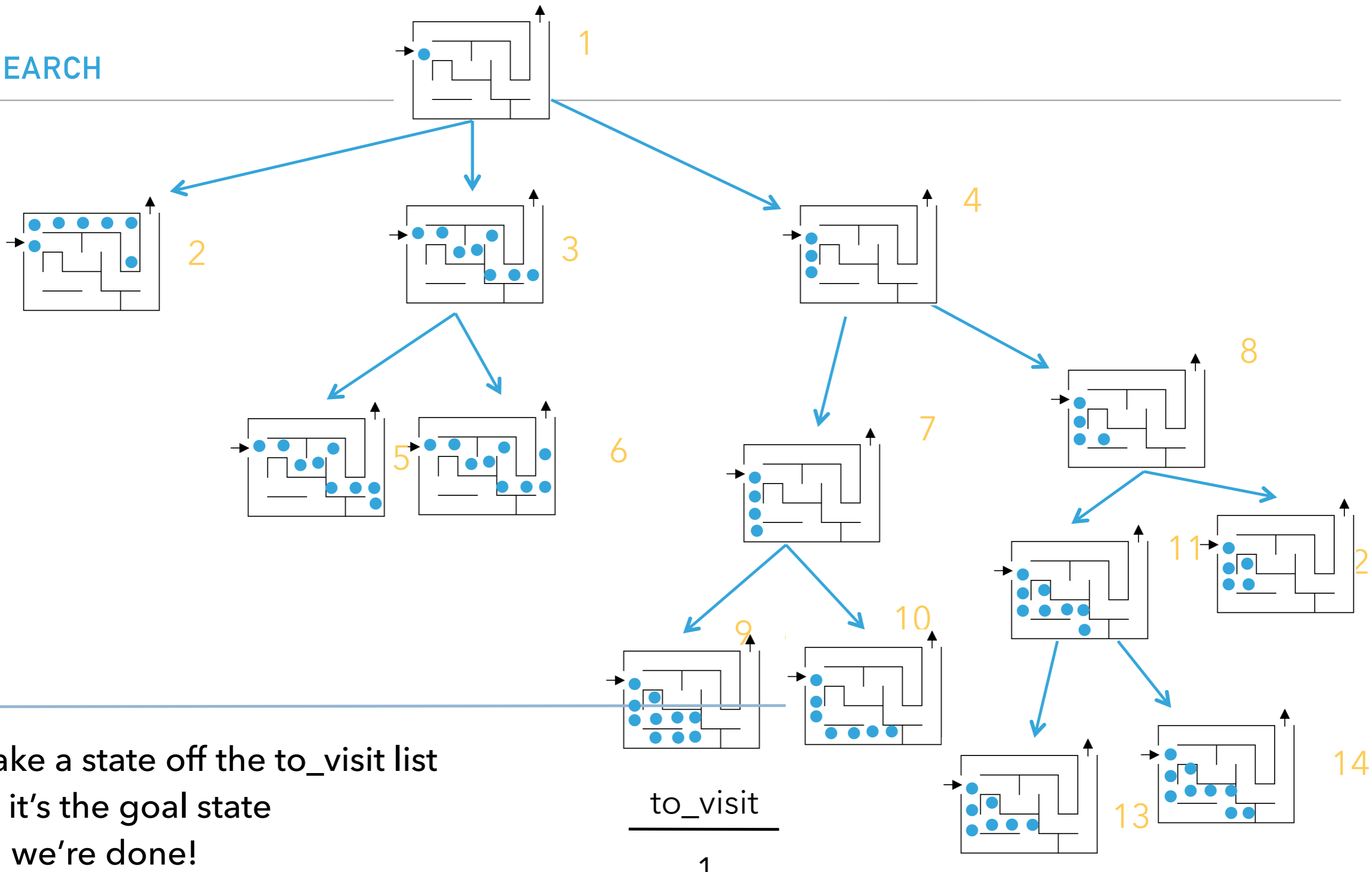


- take a state off the `to_visit` list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the `to_visit` list

- repeat

How do we start?

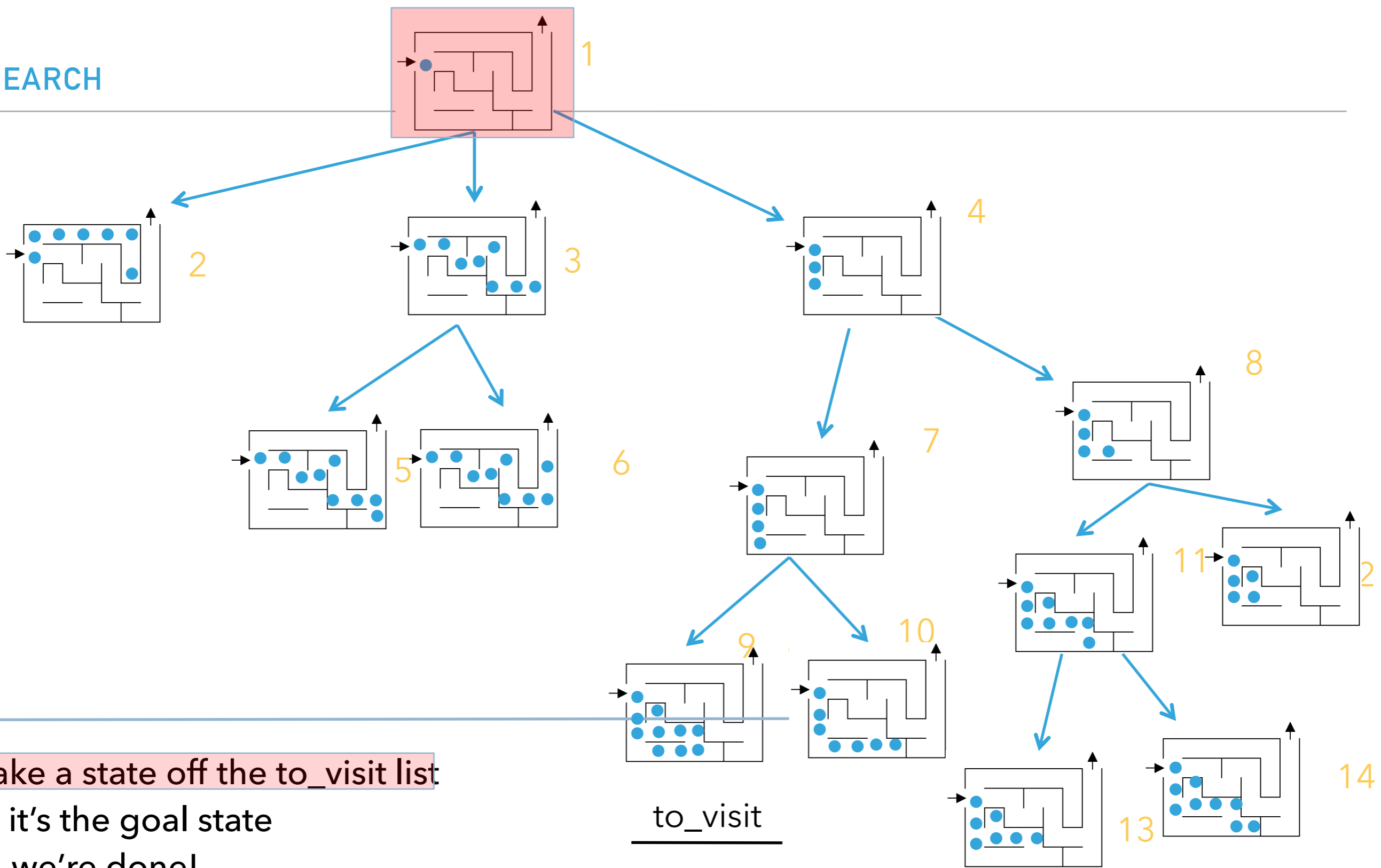
SEARCH



- take a state off the to_visit list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the to_visit list
- repeat

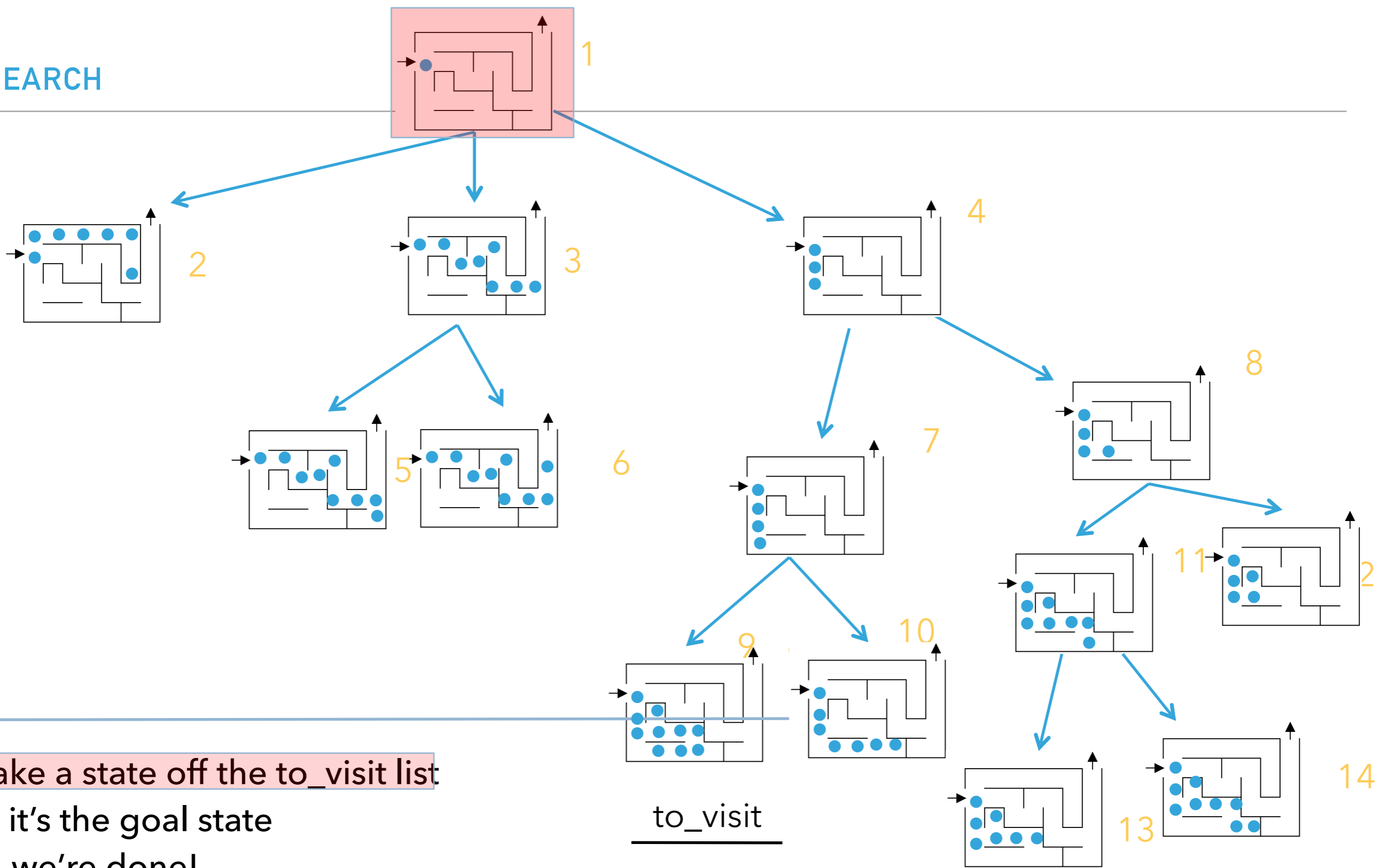
Add start to to_visit

SEARCH



- take a state off the to_visit list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the to_visit list
- repeat

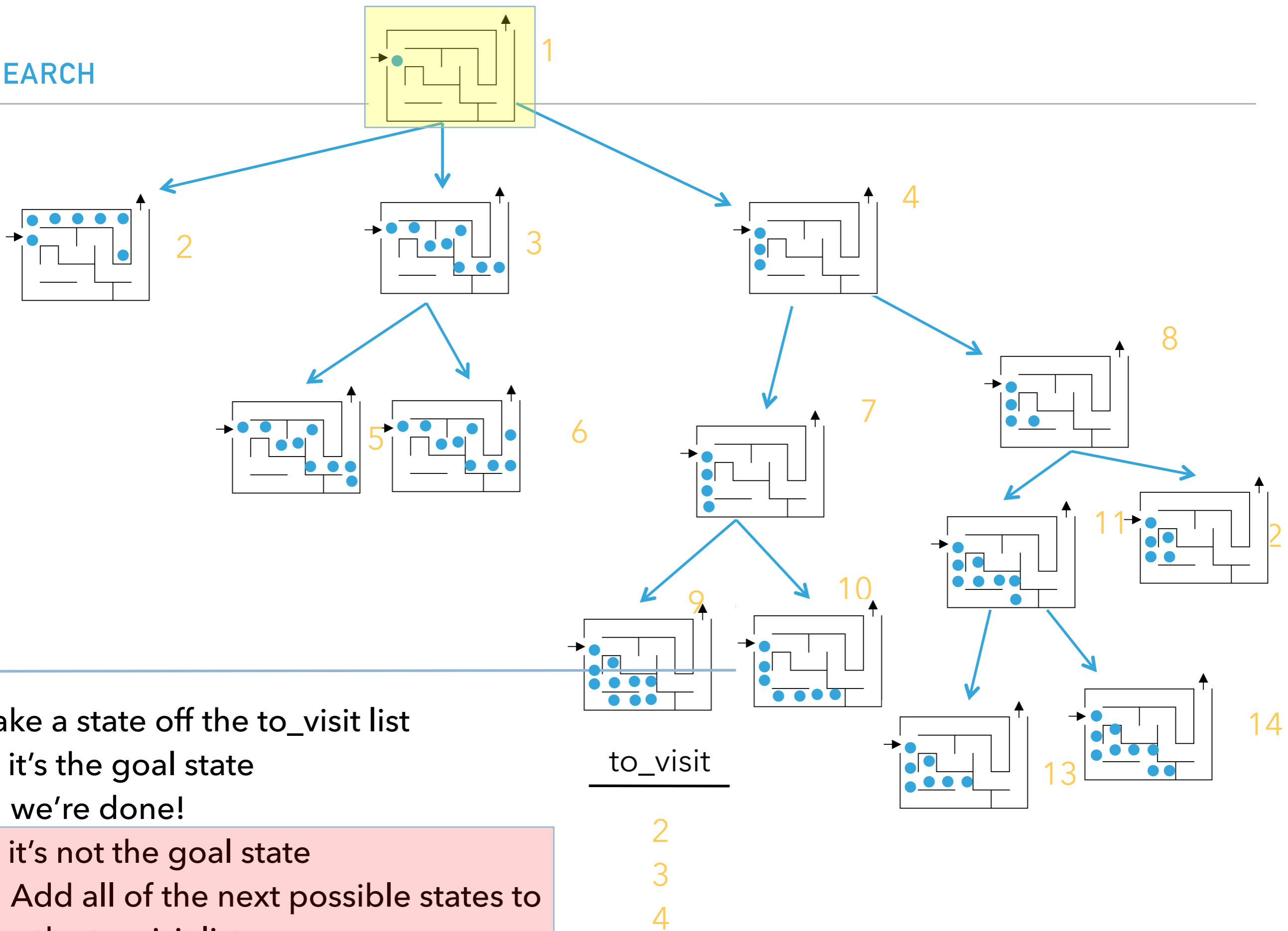
SEARCH



- take a state off the to_visit list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the to_visit list
- repeat

Is it a goal state?

SEARCH

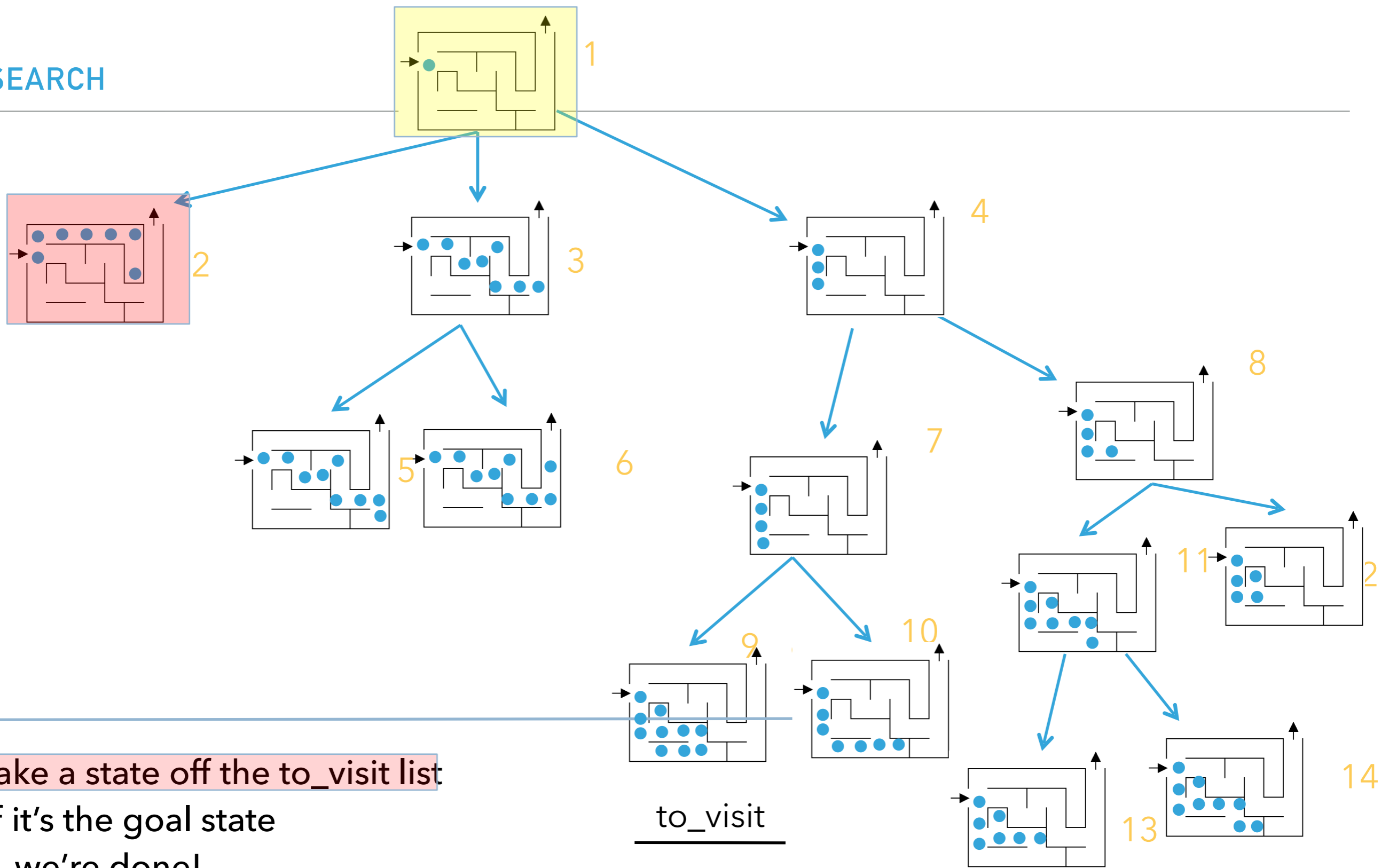


- take a state off the to_visit list
- if it's the goal state
we're done!

- if it's not the goal state
Add all of the next possible states to
the to_visit list

- repeat

SEARCH

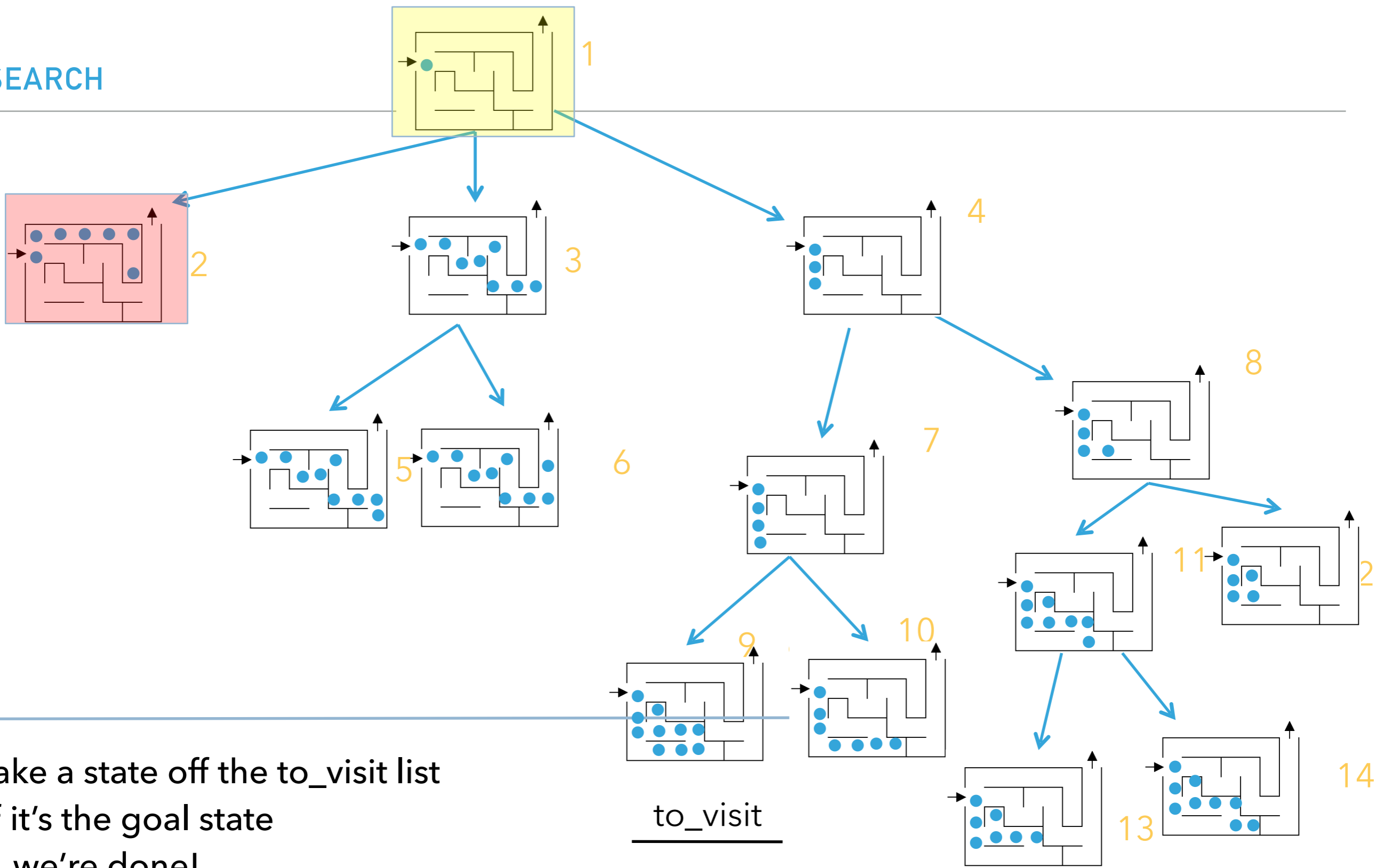


- take a state off the to_visit list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the to_visit list
- repeat

3
4

Is it a goal state?

SEARCH



- take a state off the to_visit list
- if it's the goal state we're done!

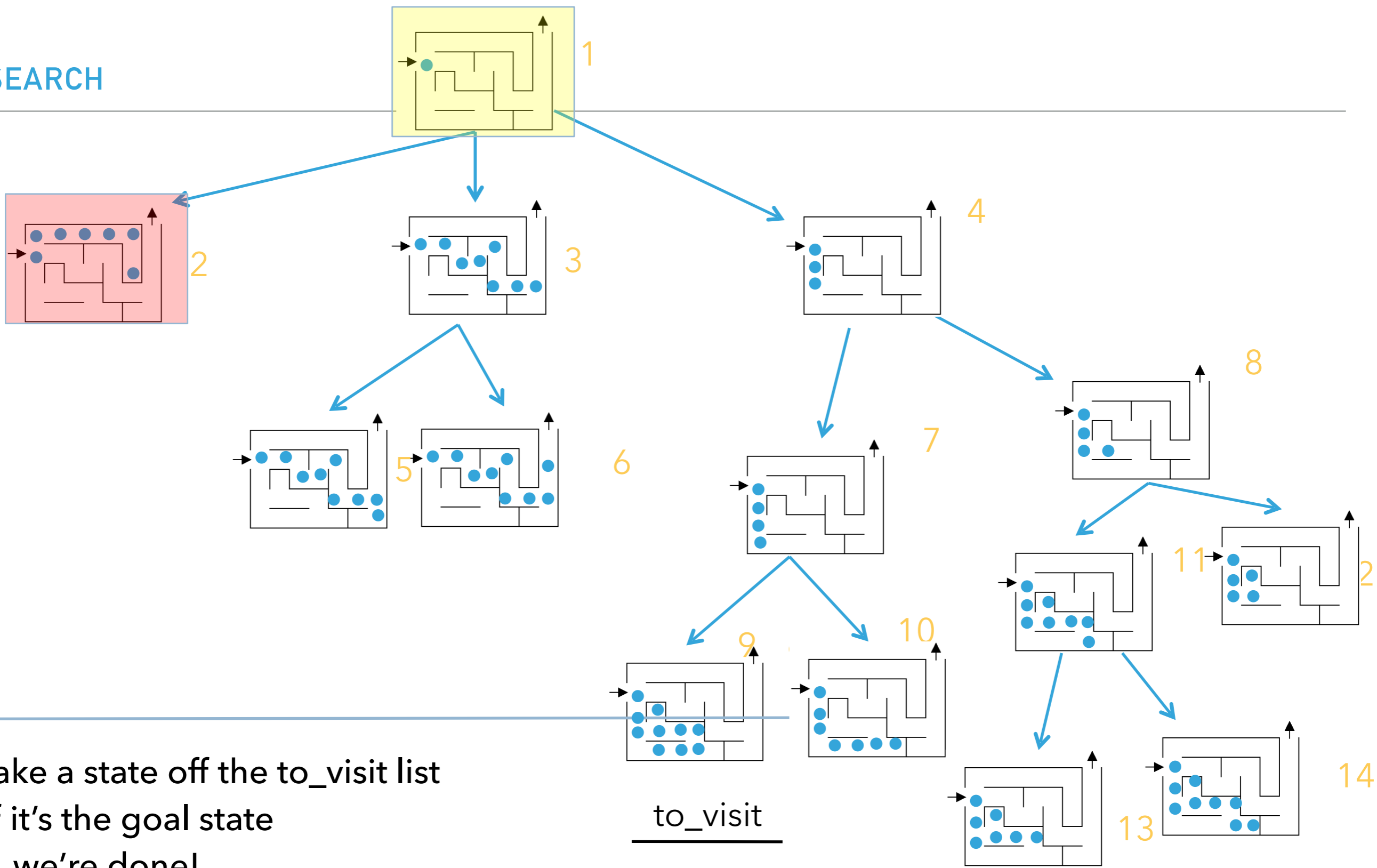
- if it's not the goal state
Add all of the next possible states to the to_visit list

- repeat

to_visit

3
4

SEARCH



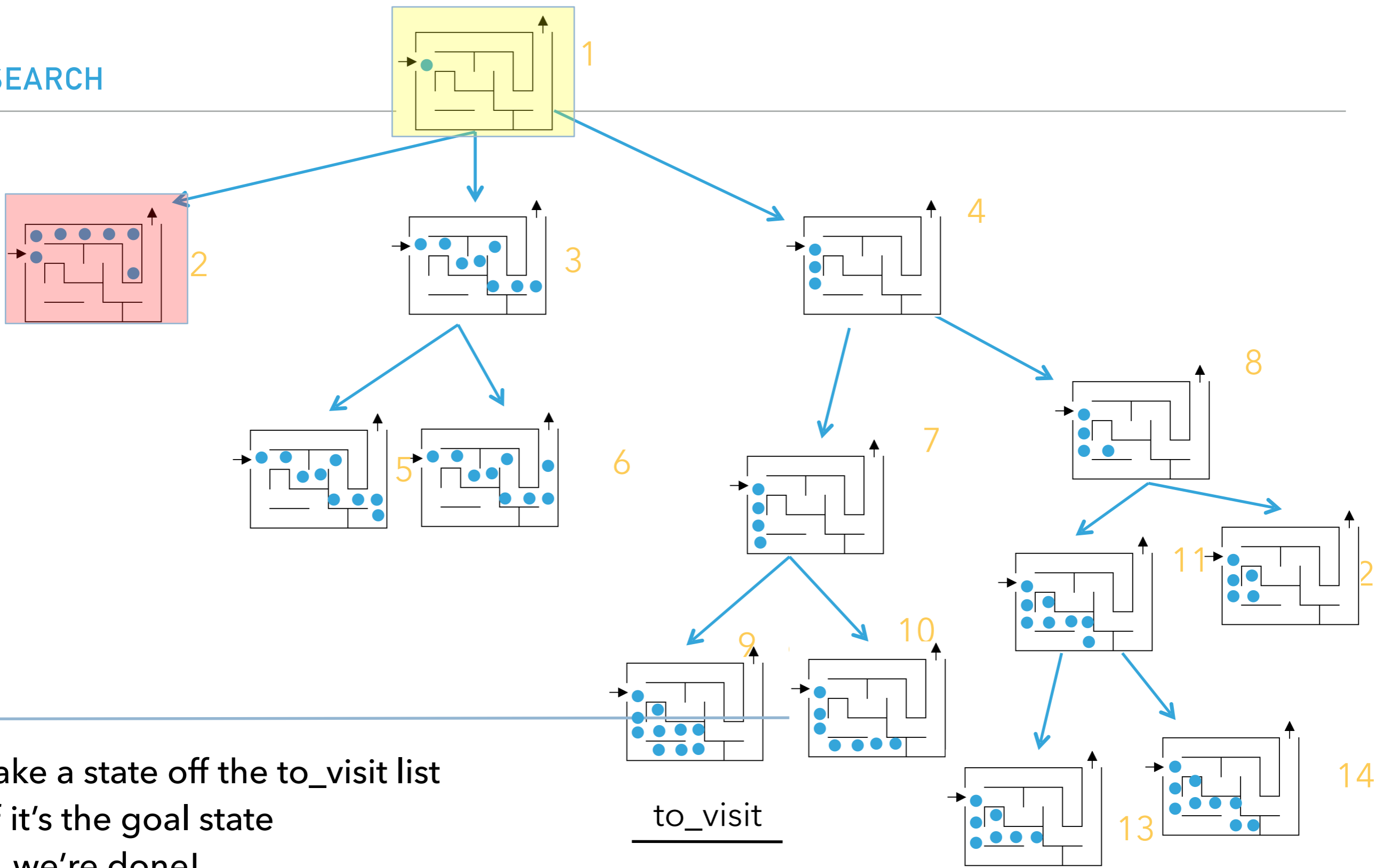
- take a state off the to_visit list
- if it's the goal state we're done!

- if it's not the goal state
Add all of the next possible states to the to_visit list

- repeat

Dead-end. What do we do now?

SEARCH



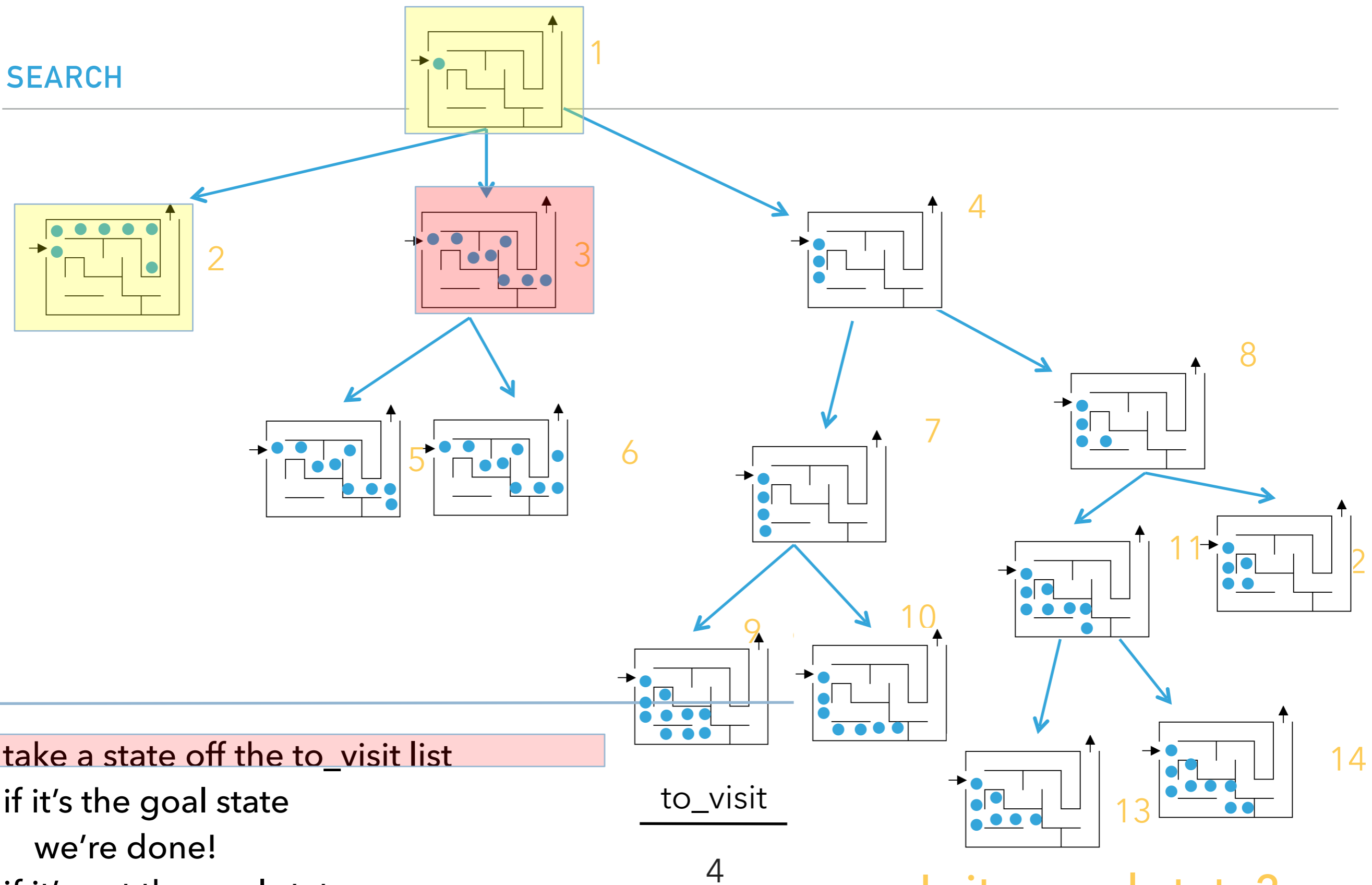
- take a state off the to_visit list
- if it's the goal state we're done!

- if it's not the goal state
Add all of the next possible states to the to_visit list

- repeat

to_visit list keeps track of where to go next, i.e. the states we know about but haven't explored

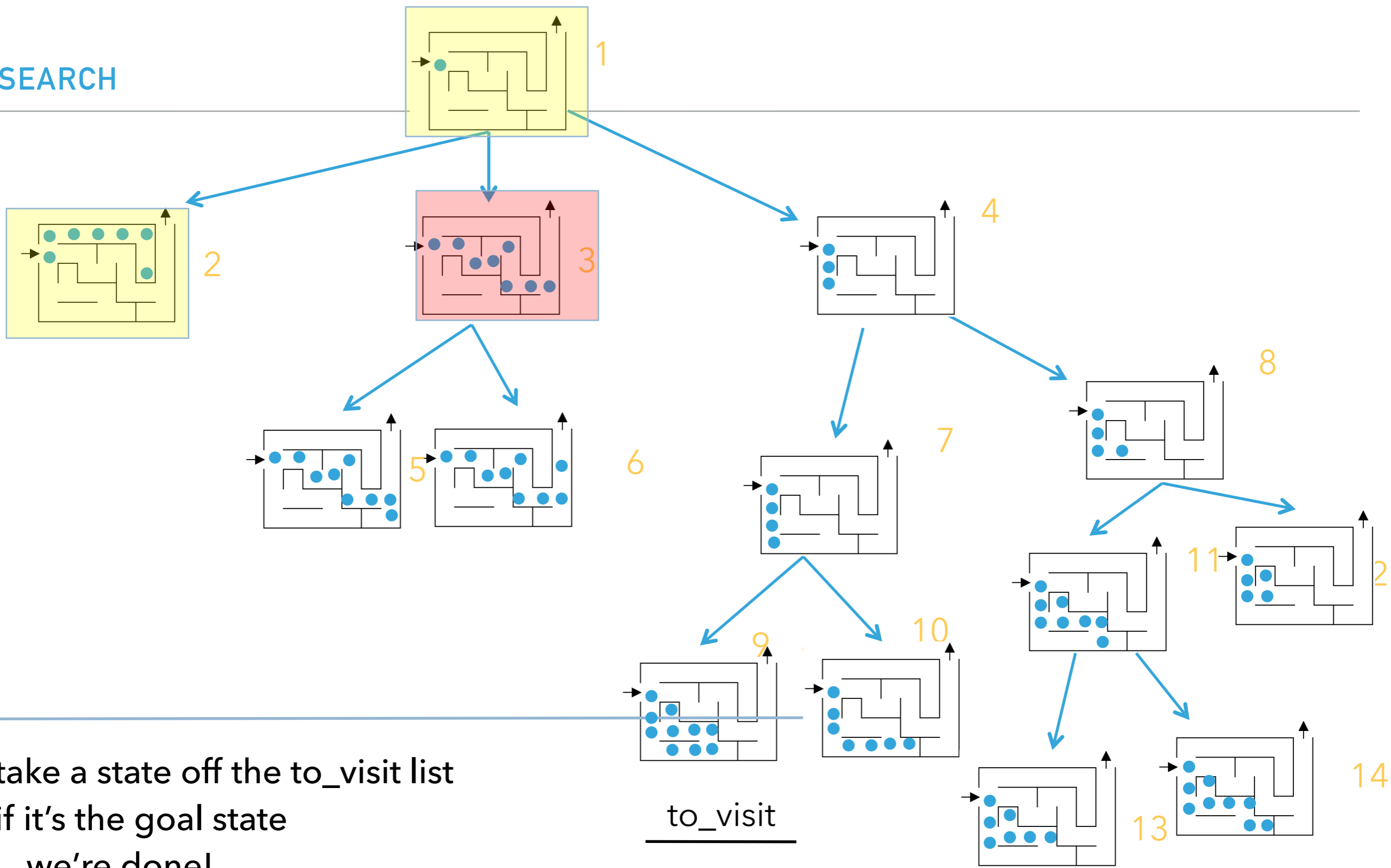
SEARCH



- take a state off the to_visit list
- if it's the goal state we're done!
- if it's not the goal state Add all of the next possible states to the to_visit list
- repeat

Is it a goal state?

SEARCH



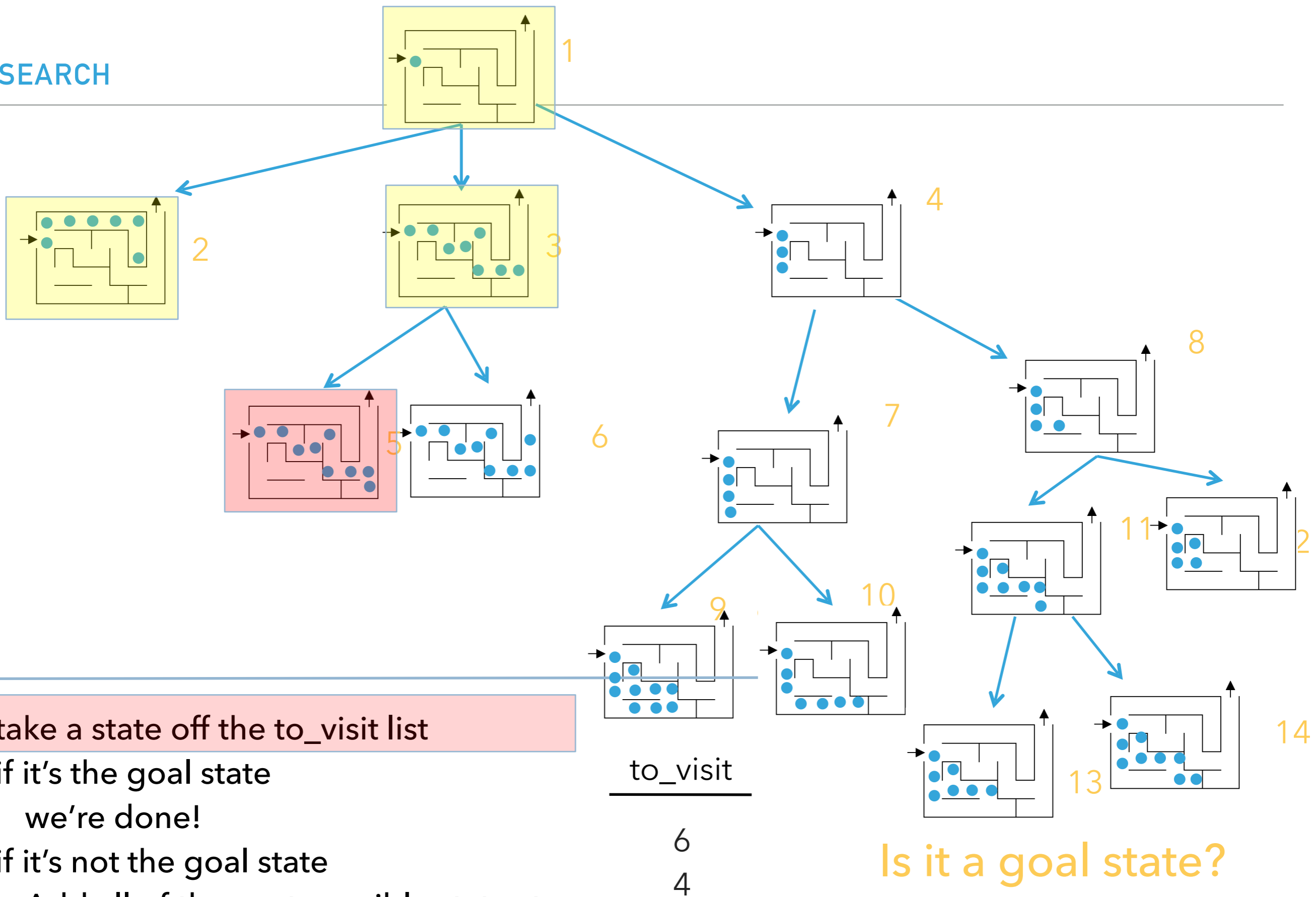
- take a state off the to_visit list
- if it's the goal state we're done!

- if it's not the goal state
Add all of the next possible states to the to_visit list

- repeat

5
6
4

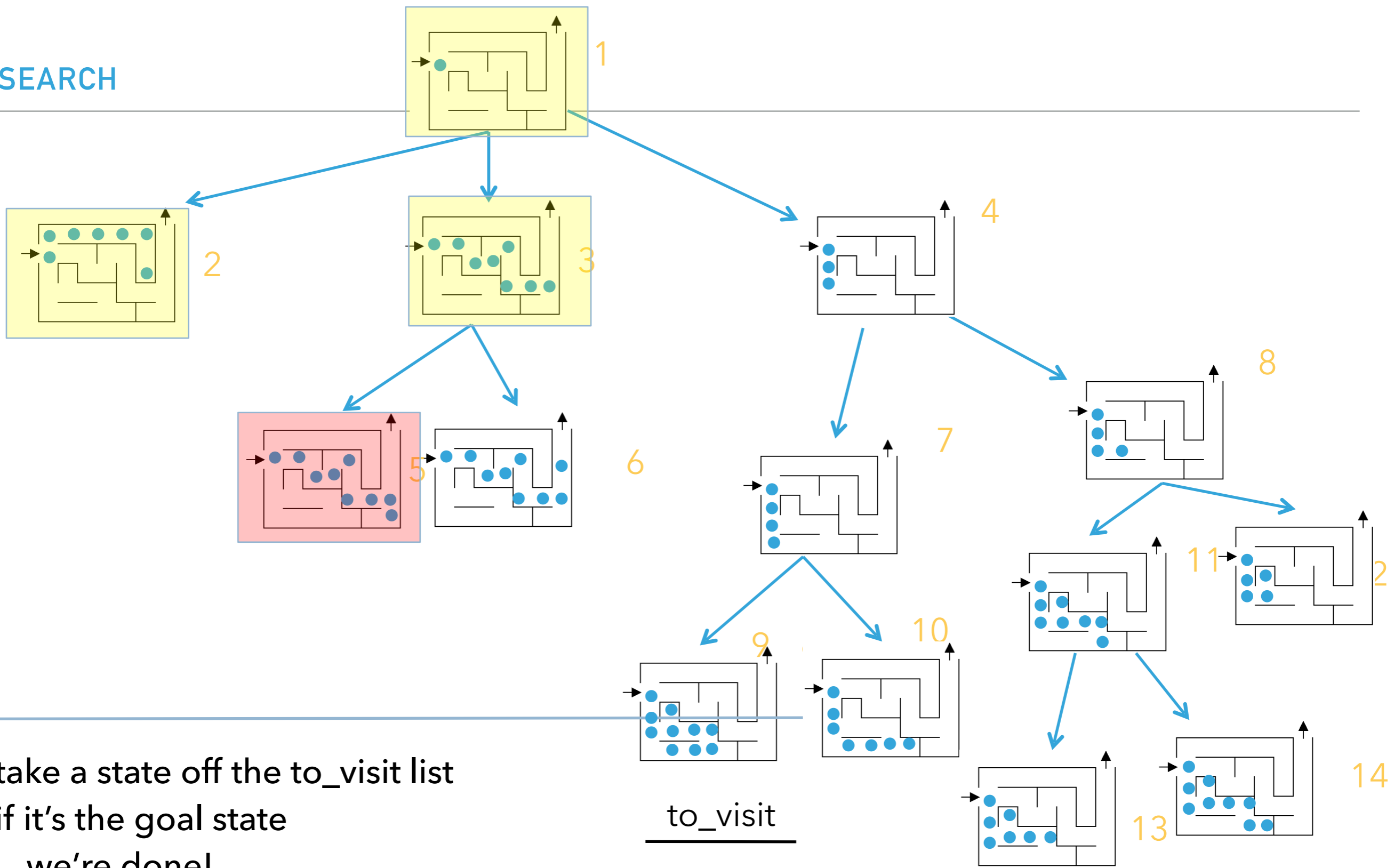
SEARCH



- take a state off the to_visit list
- if it's the goal state we're done!
- if it's not the goal state Add all of the next possible states to the to_visit list
- repeat

Is it a goal state?

SEARCH

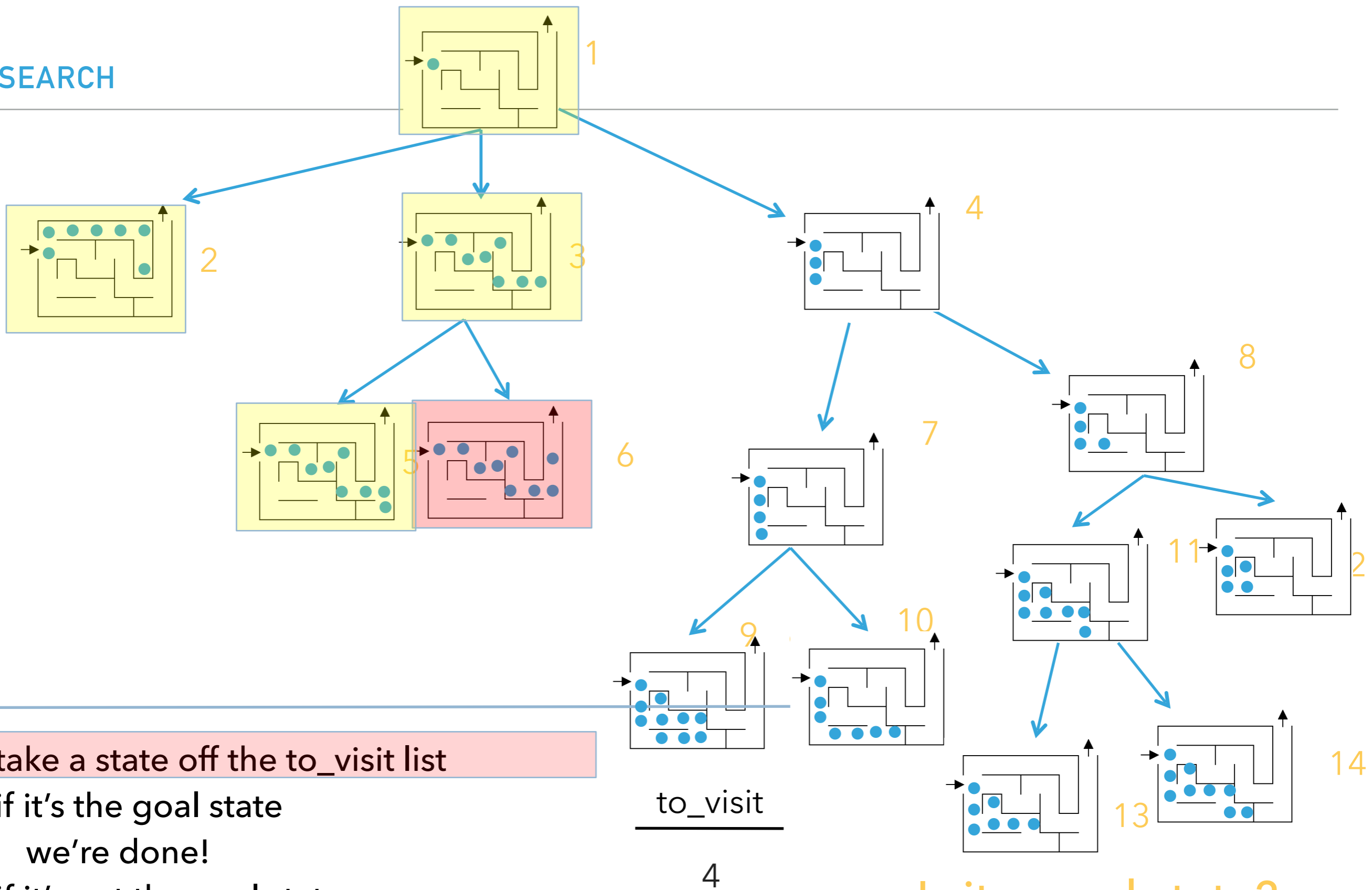


- take a state off the to_visit list
- if it's the goal state
we're done!

- if it's not the goal state
Add all of the next possible states to
the to_visit list

- repeat

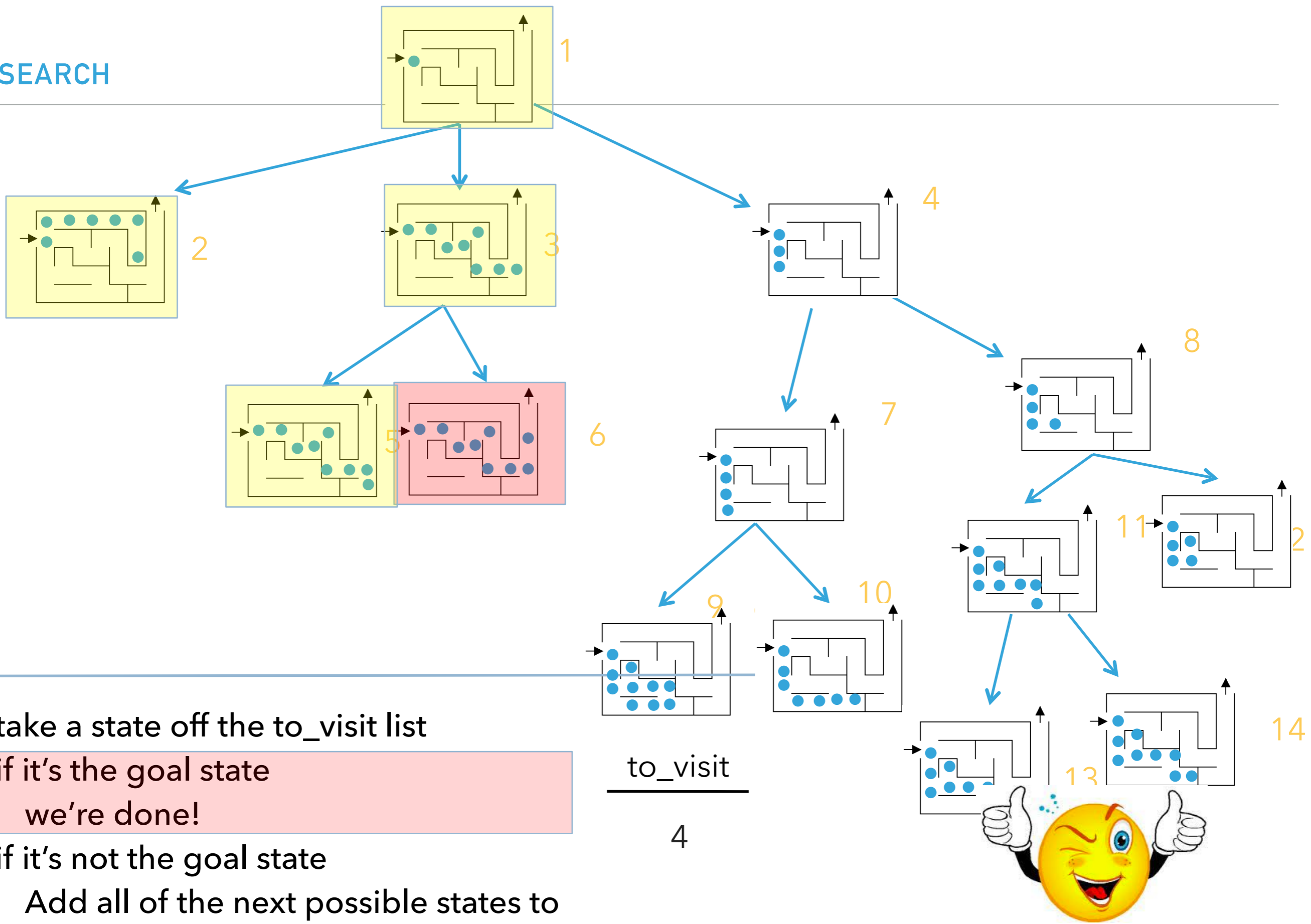
SEARCH



- take a state off the `to_visit` list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the `to_visit` list
- repeat

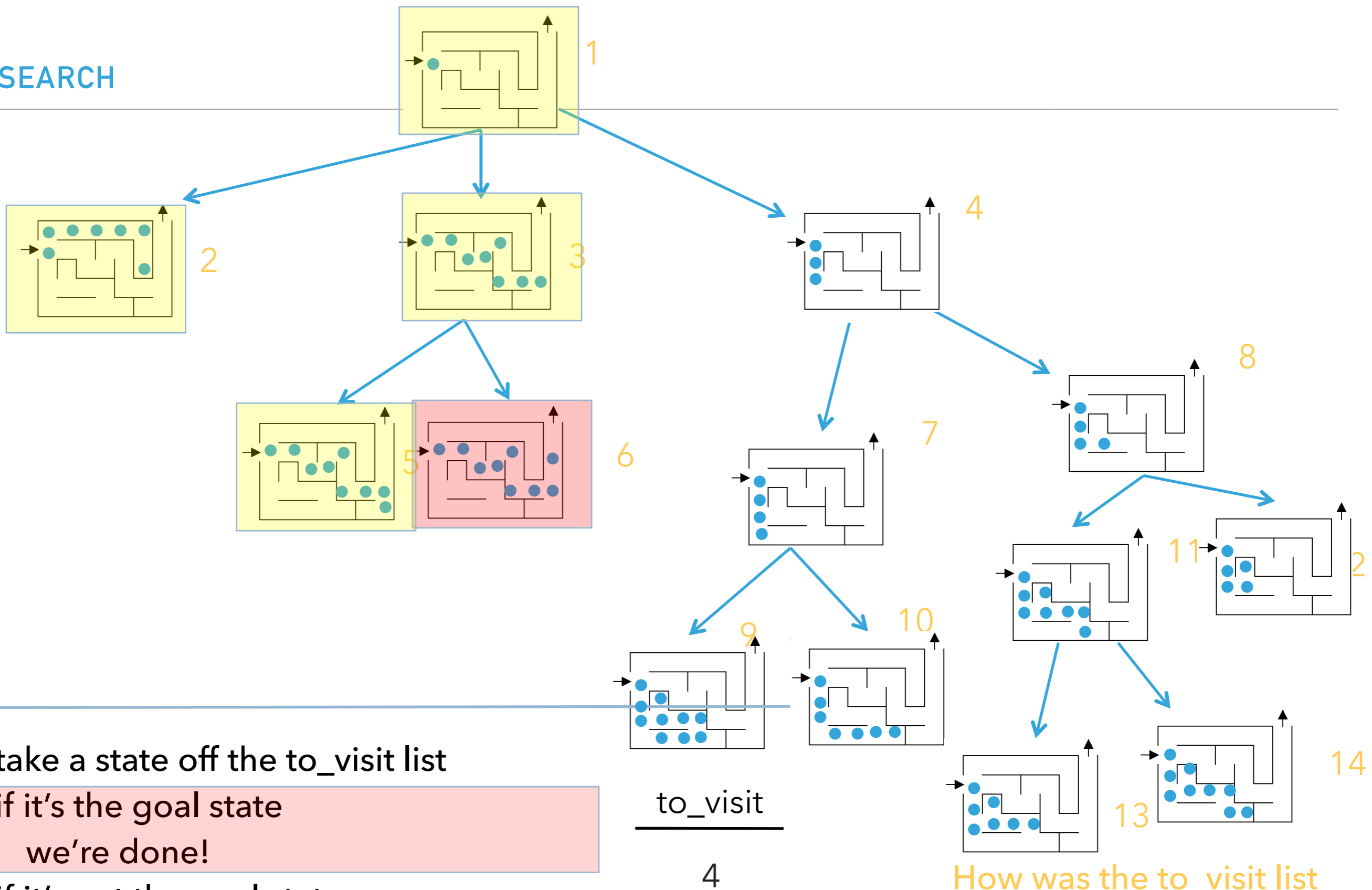
Is it a goal state?

SEARCH



- take a state off the to_visit list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the to_visit list
- repeat

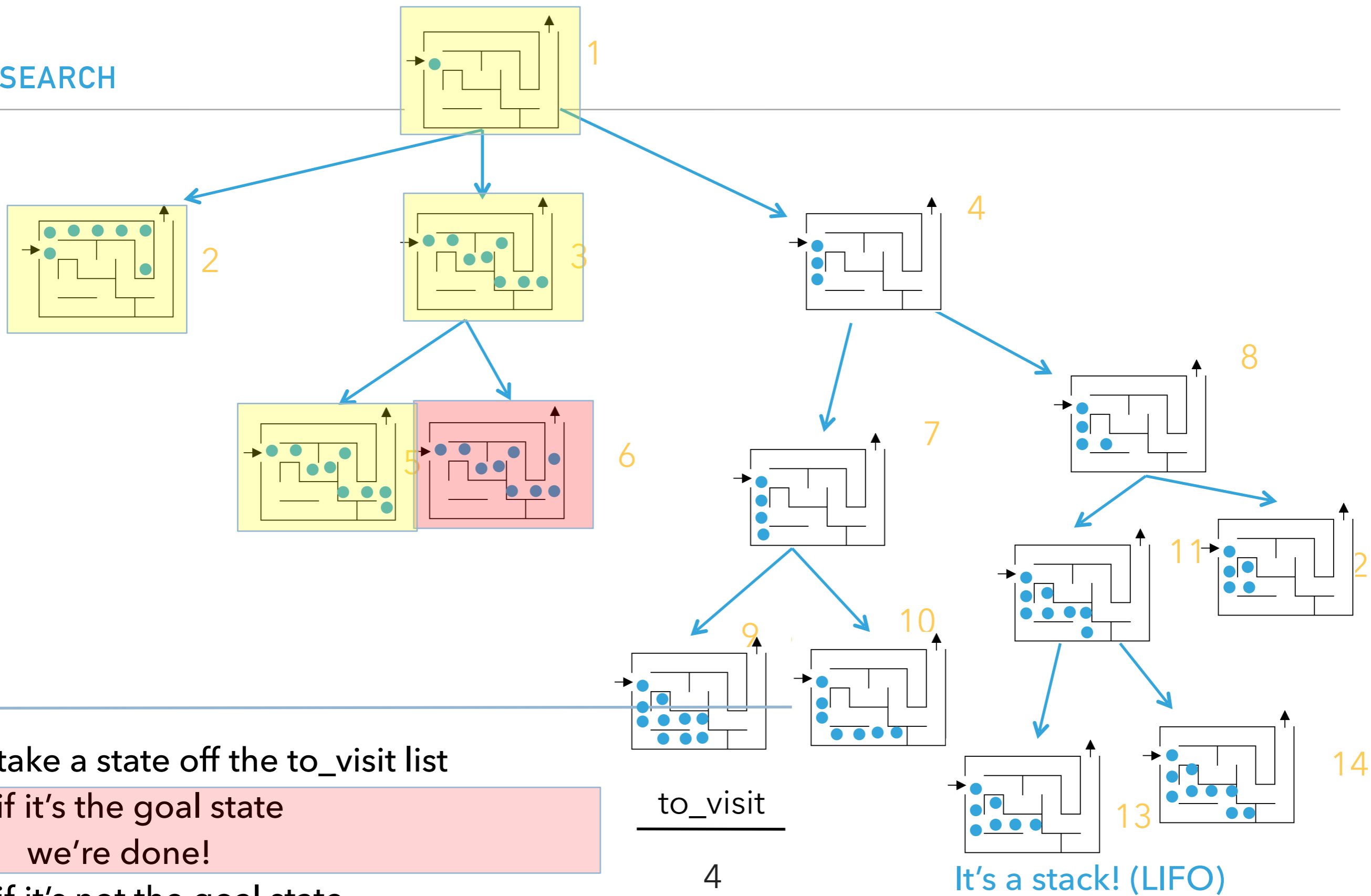
SEARCH



- take a state off the to_visit list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the to_visit list
- repeat

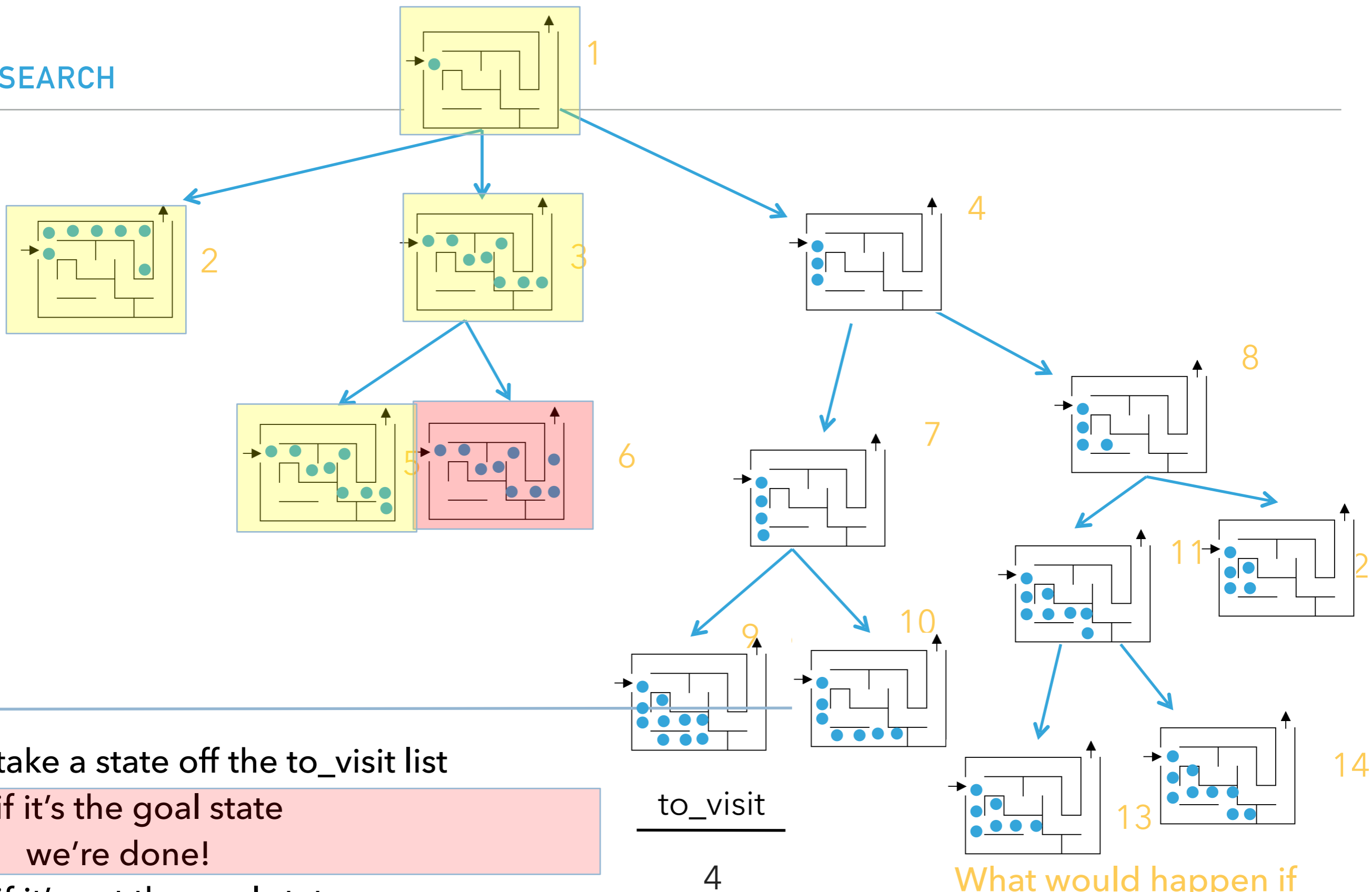
How was the to_visit list organized in this example, i.e., what order?

SEARCH



- take a state off the to_visit list
- if it's the goal state
we're done!
- if it's not the goal state
Add all of the next possible states to
the to_visit list
- repeat

SEARCH



- take a state off the to_visit list
- if it's the goal state we're done!
- if it's not the goal state
Add all of the next possible states to the to_visit list
- repeat

What would happen if we used a queue?

Search algorithms

- ▶ add the start state to to_visit
- ▶ Repeat
 - ▶ take a state off the to_visit list
 - ▶ if it's the goal state
 - ▶ we're done!
 - ▶ if it's not the goal state
 - ▶ Add all of the next possible states to the to_visit list

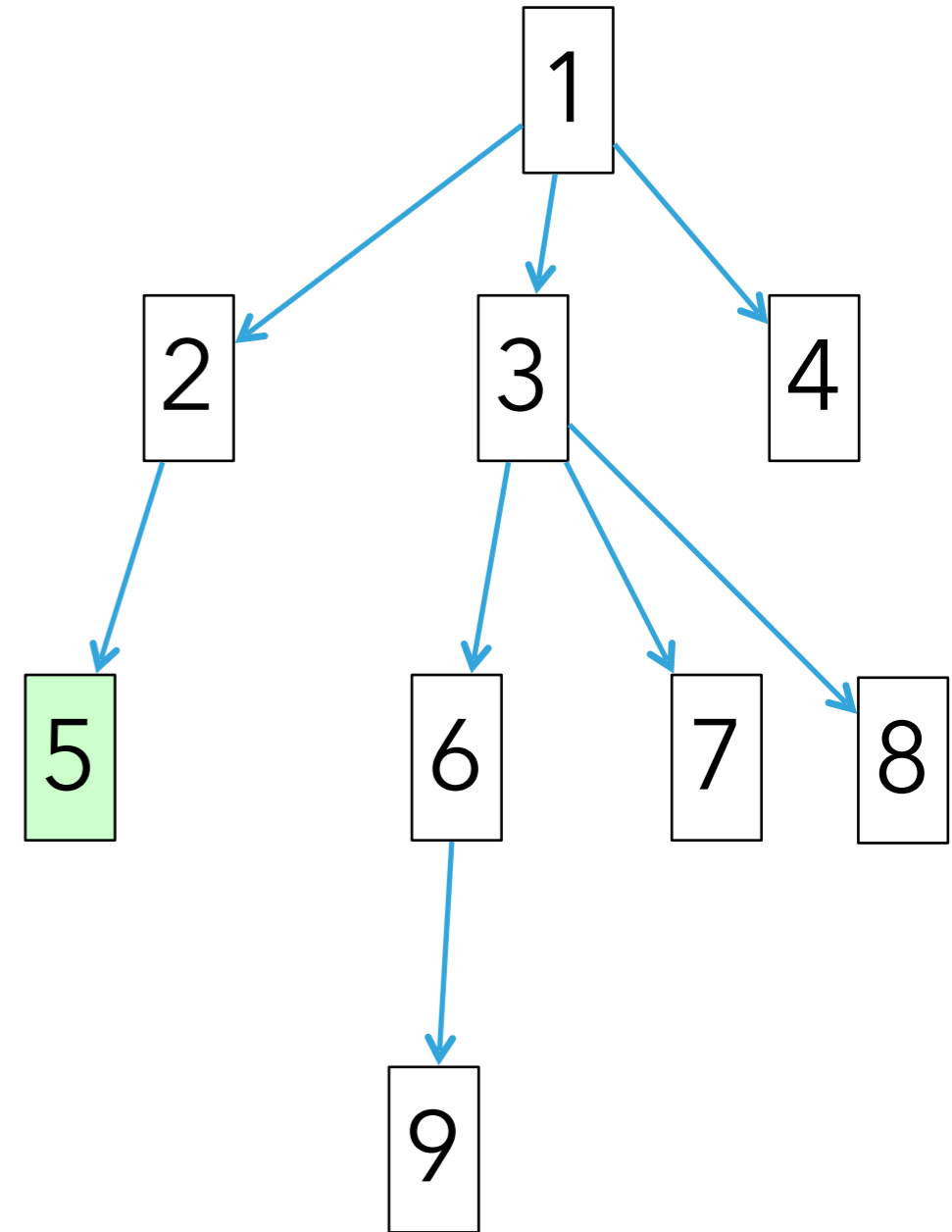
Search algorithms

- ▶ add the start state to to_visit
- ▶ Repeat
 - ▶ take a state off the to_visit list
 - ▶ if it's the goal state
 - ▶ we're done!
 - ▶ if it's not the goal state
 - ▶ Add all of the next possible states to the to_visit list
- ▶ Depth first search (DFS): to_visit is a stack
- ▶ Breadth first search (BFS): to_visit is a queue

SEARCH

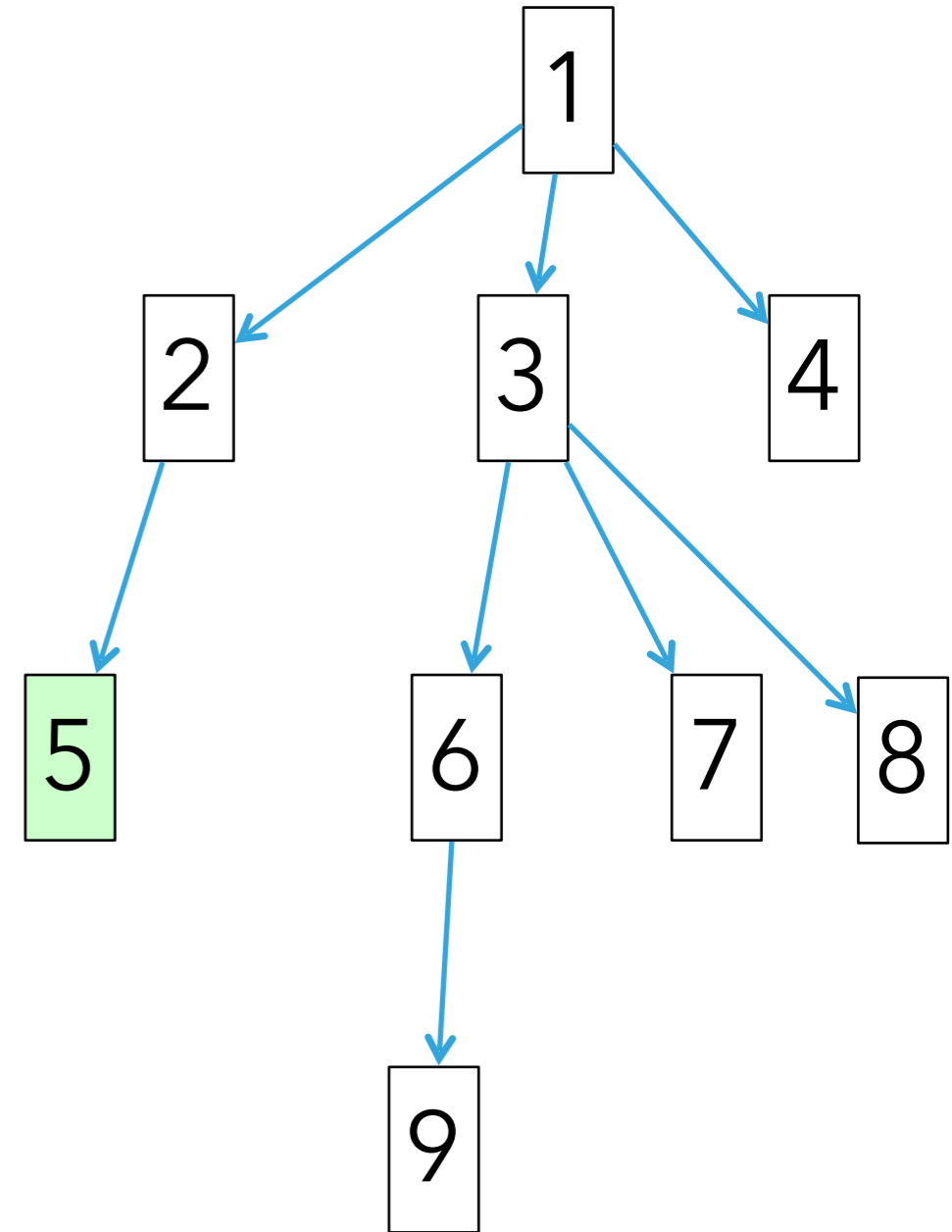
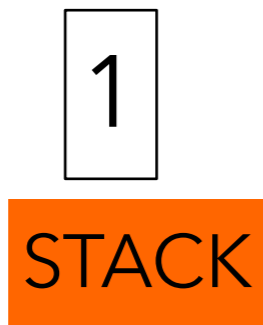
What order will BFS and DFS visit the states assuming states are added to to_visit left to right?

- ▶ add the start state to to_visit
- ▶ Repeat
 - ▶ take a state off the to_visit list
 - ▶ if it's the goal state
 - ▶ we're done!
 - ▶ if it's not the goal state
 - ▶ Add all of the next states to the to_visit list
- ▶ Depth first search (DFS): to_visit is a stack
- ▶ Breadth first search (BFS): to_visit is a queue



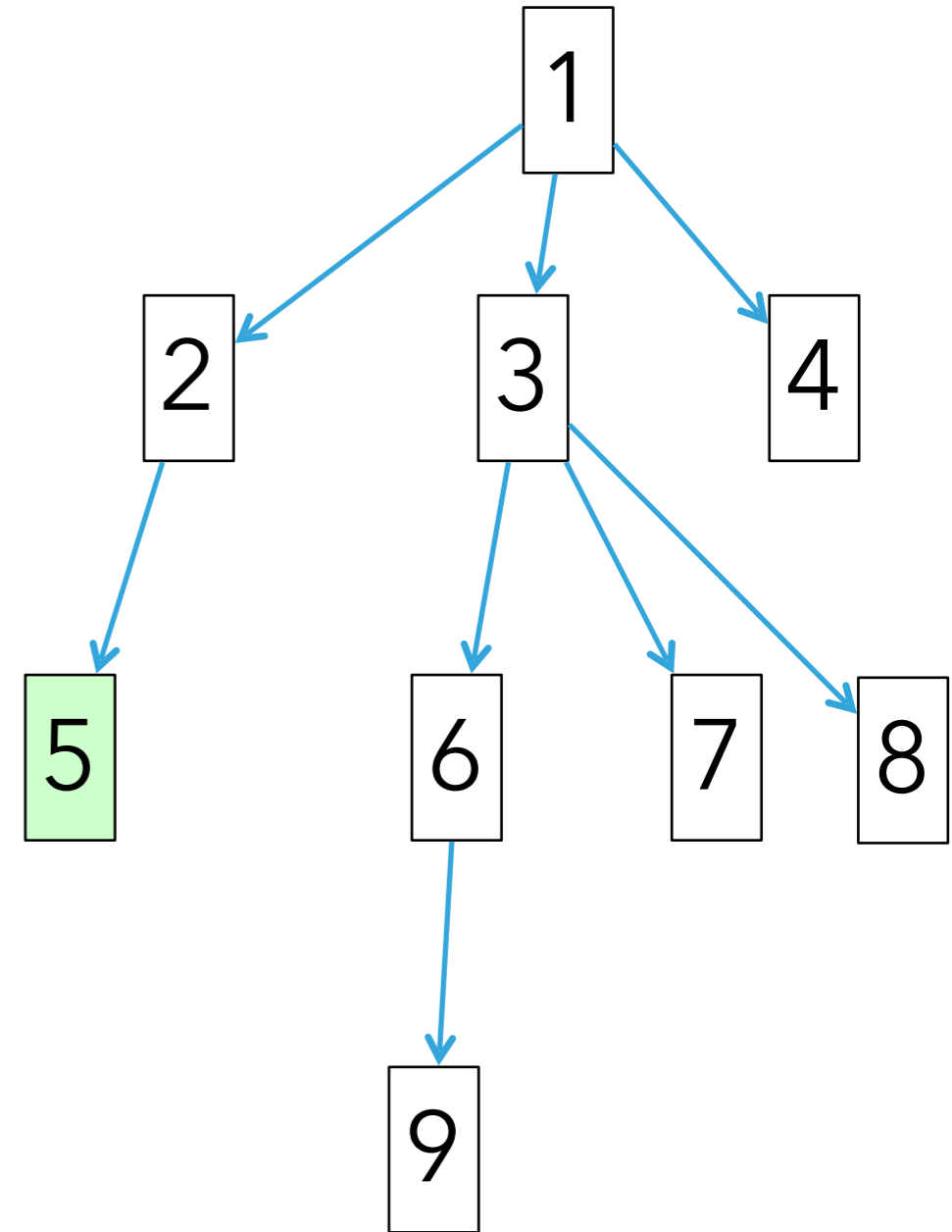
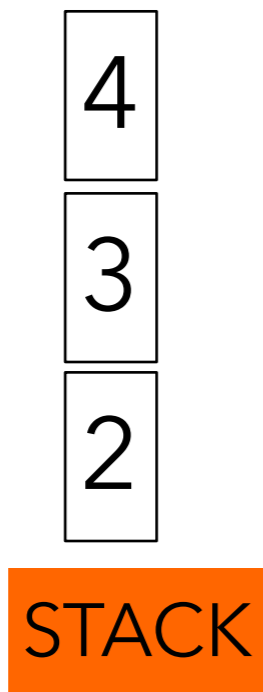
What order will BFS and DFS visit the states?

▶ DFS:



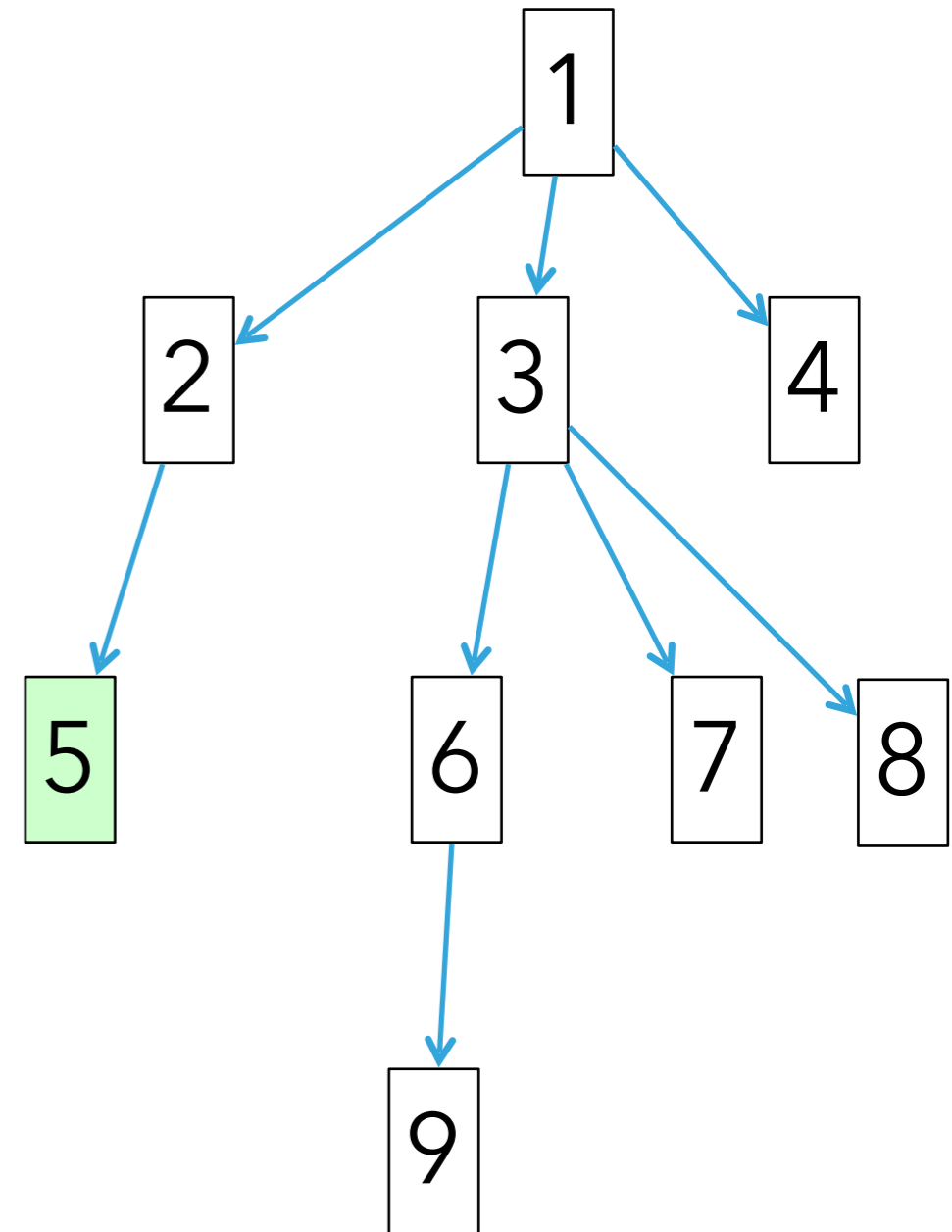
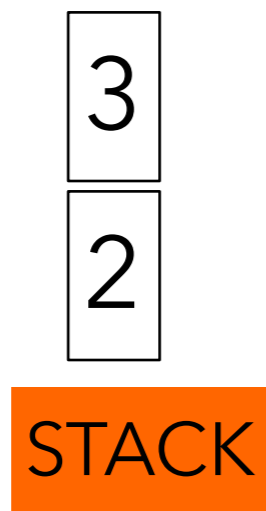
What order will BFS and DFS visit the states?

- ▶ DFS: 1



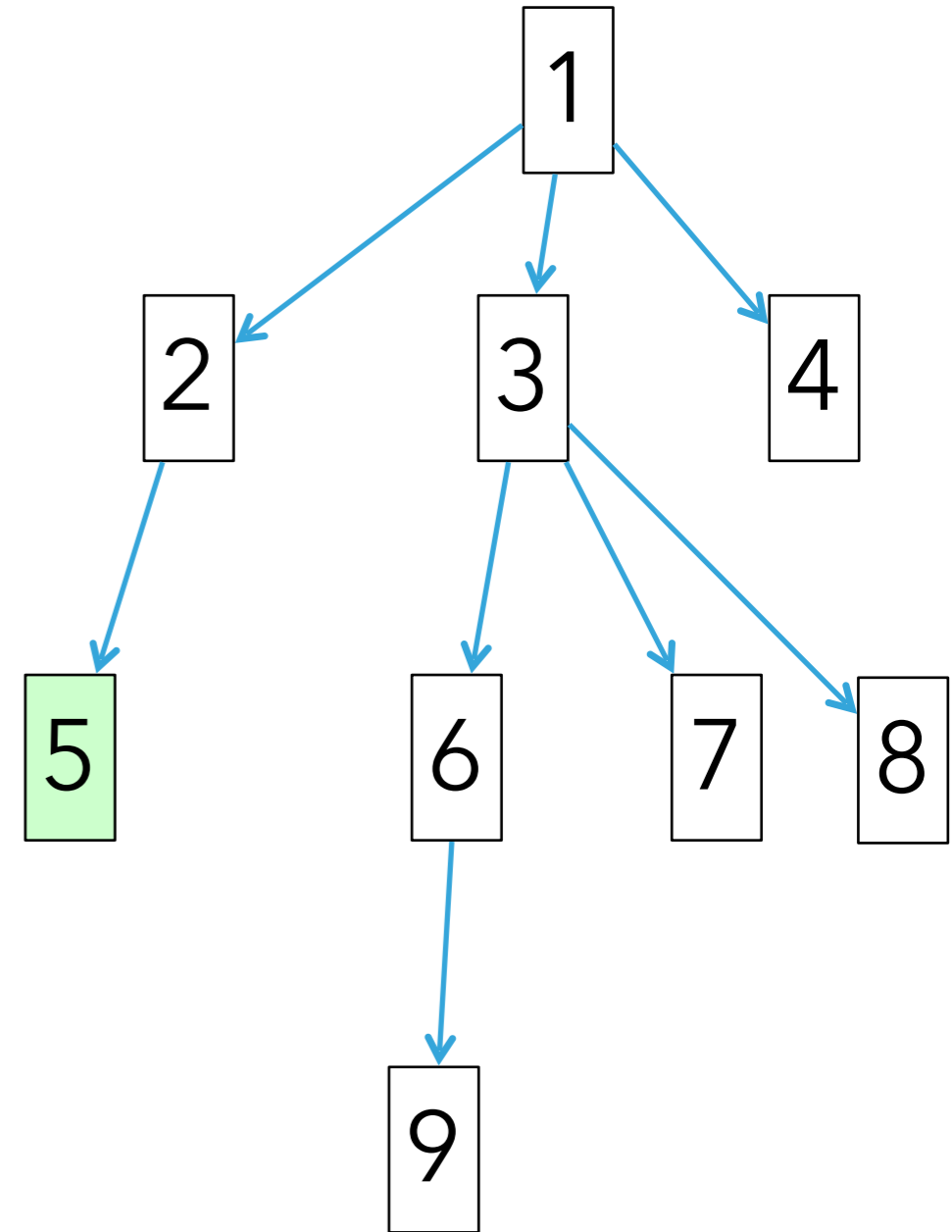
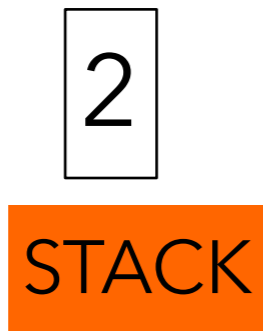
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4



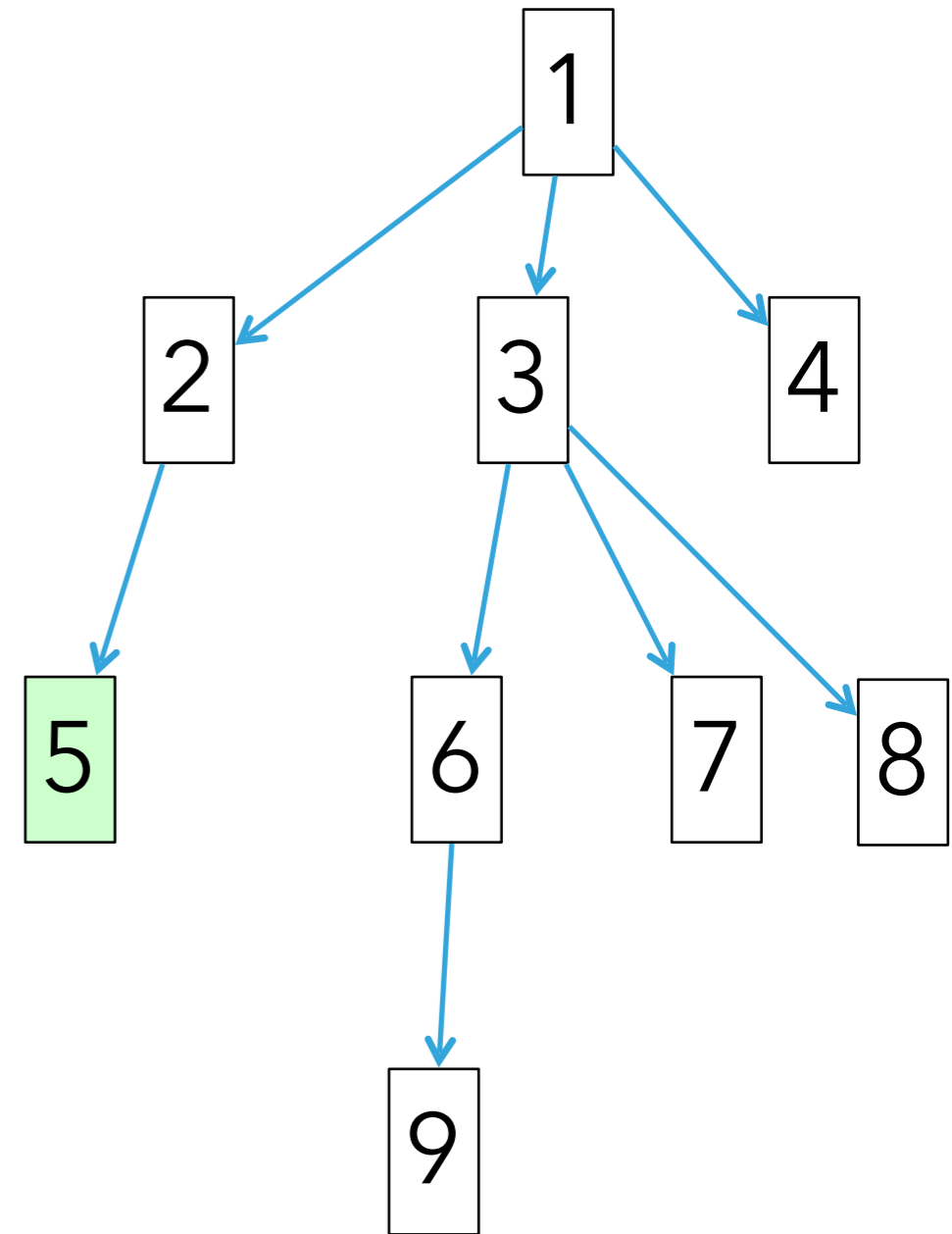
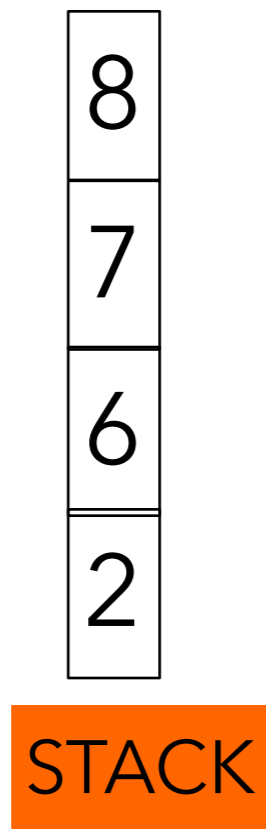
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3



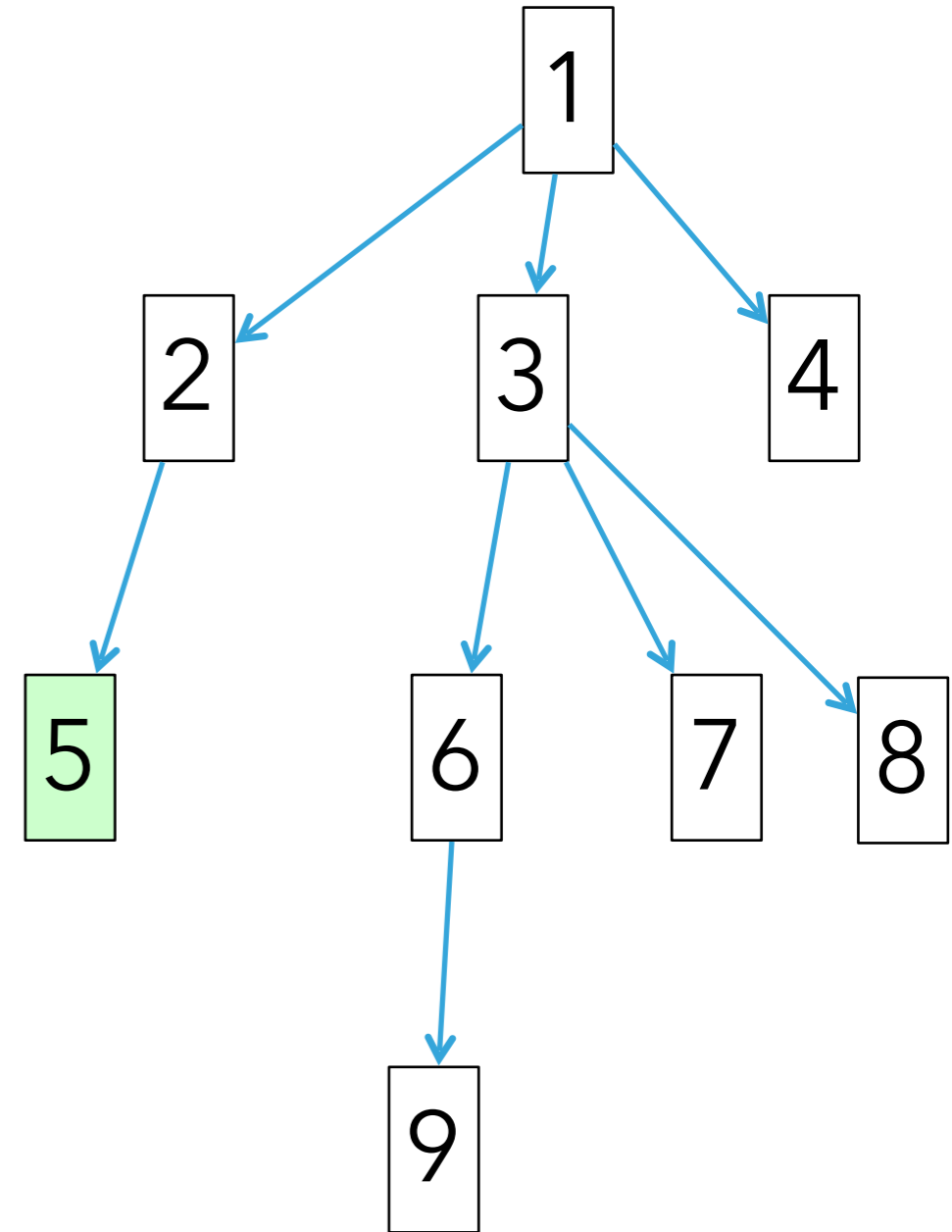
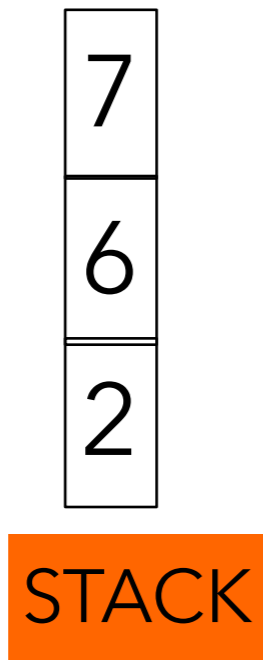
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3



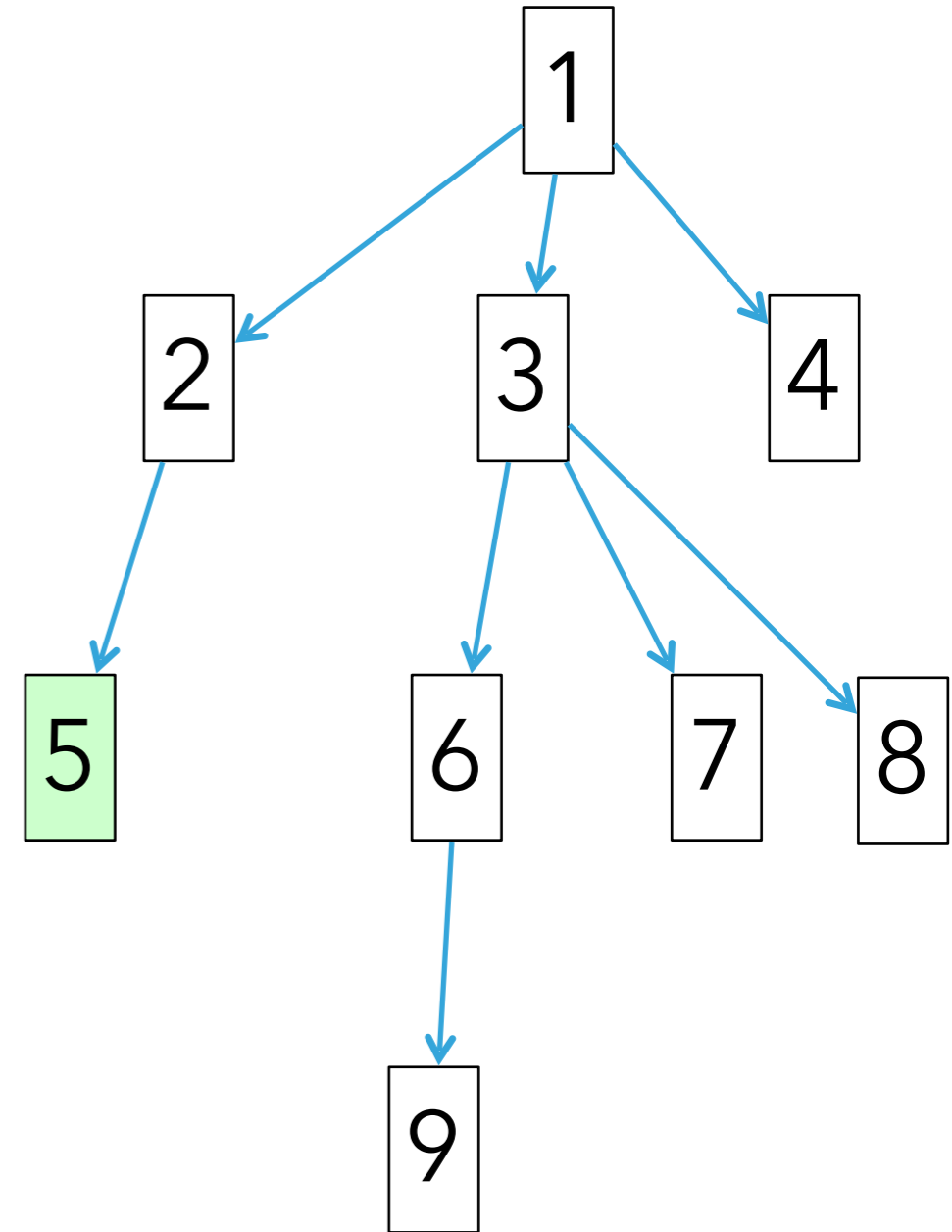
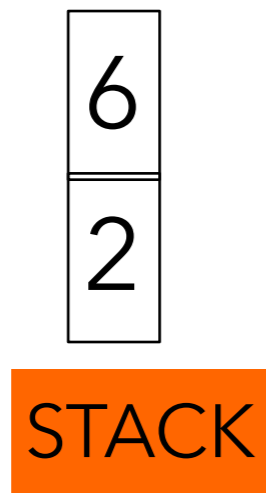
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8



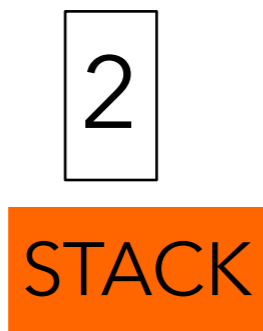
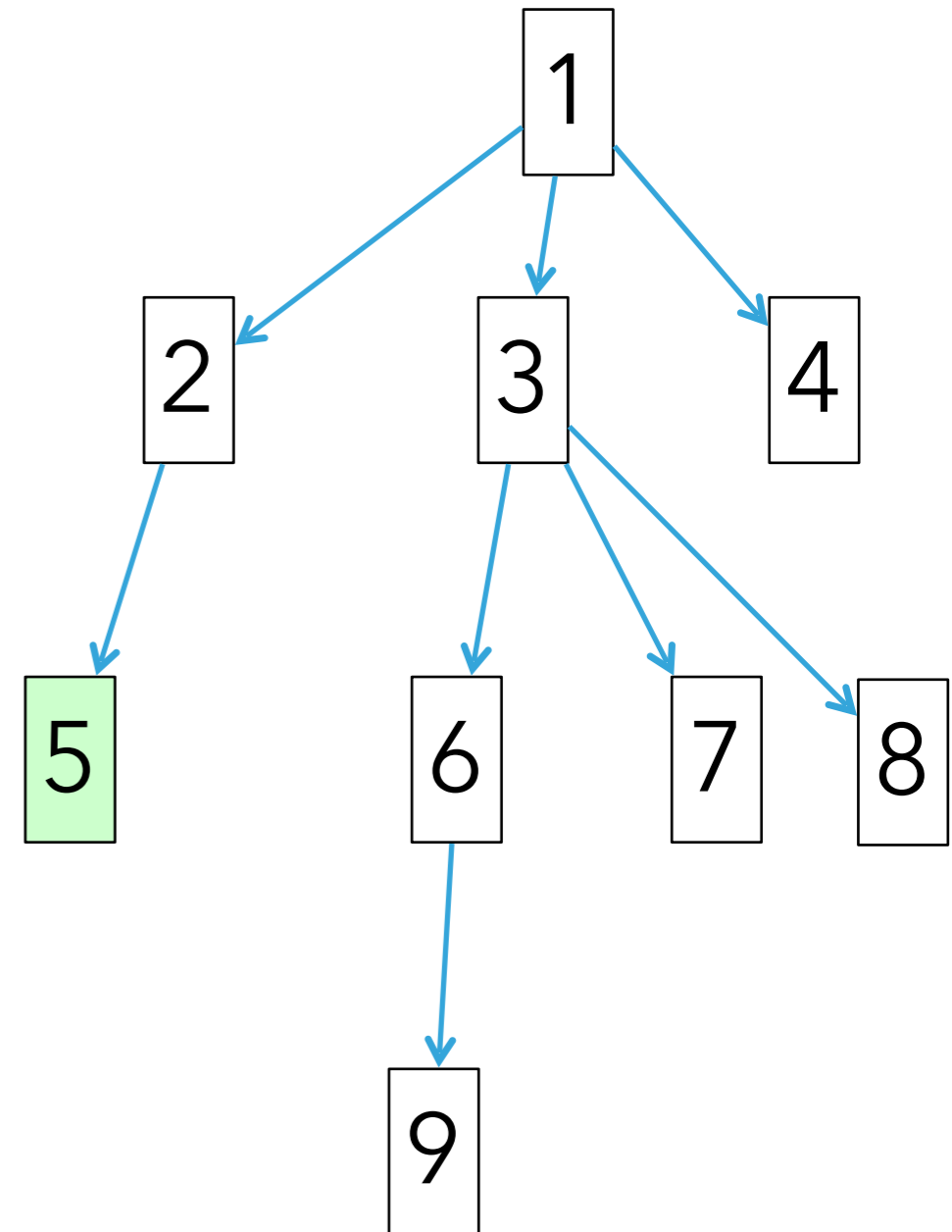
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7



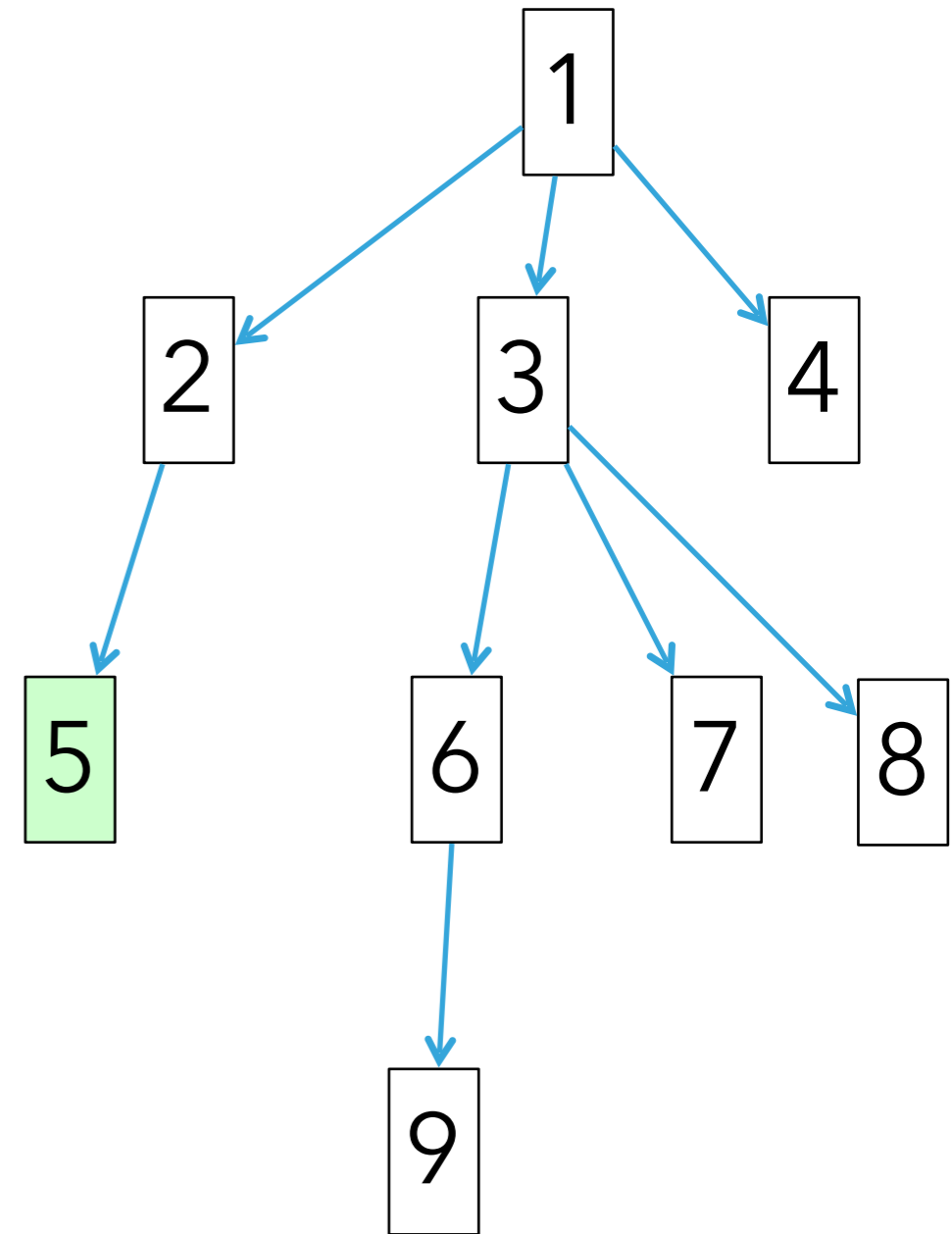
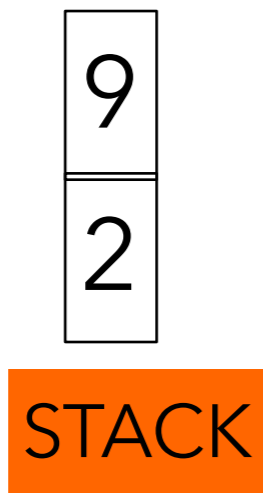
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6



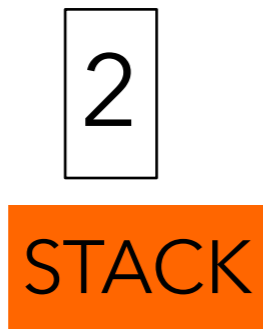
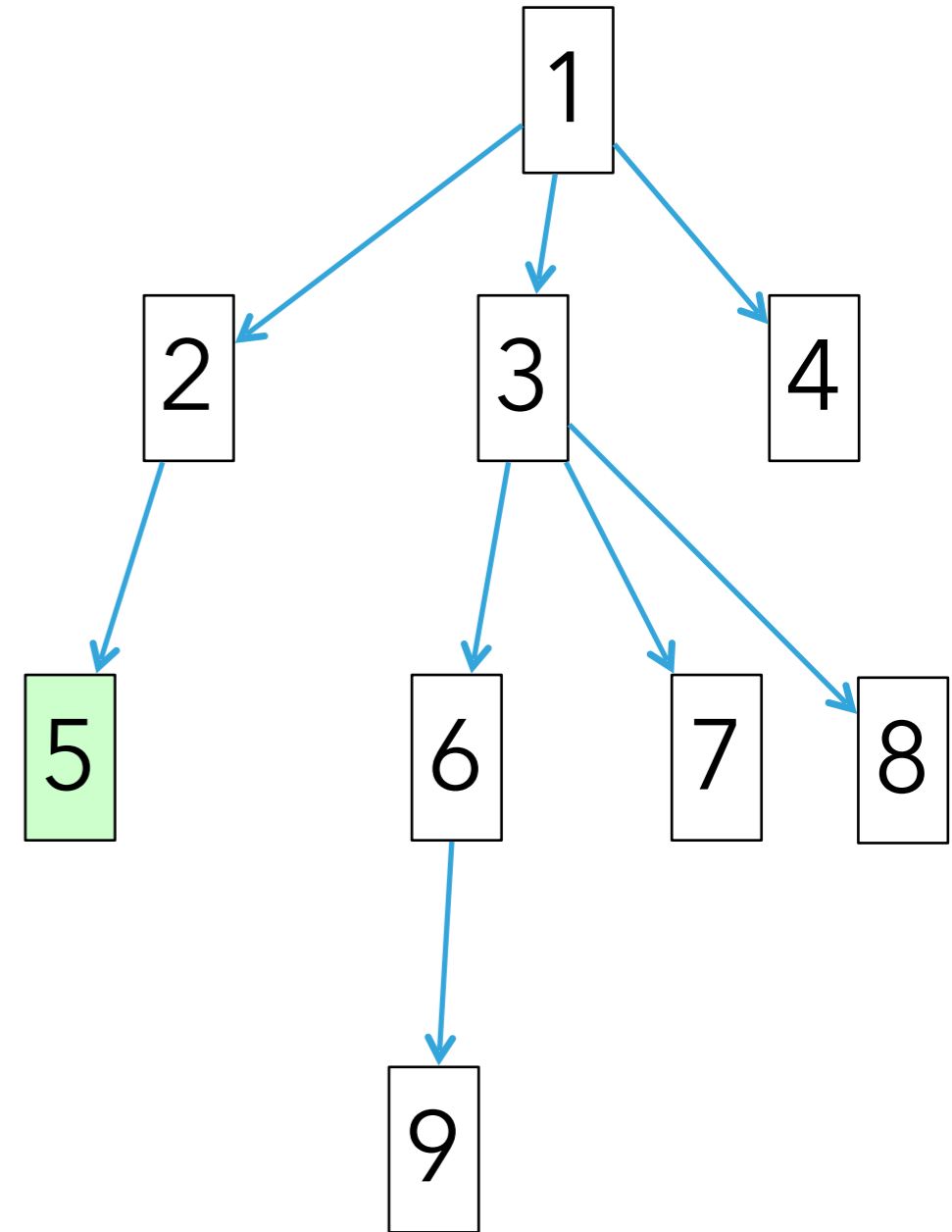
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6



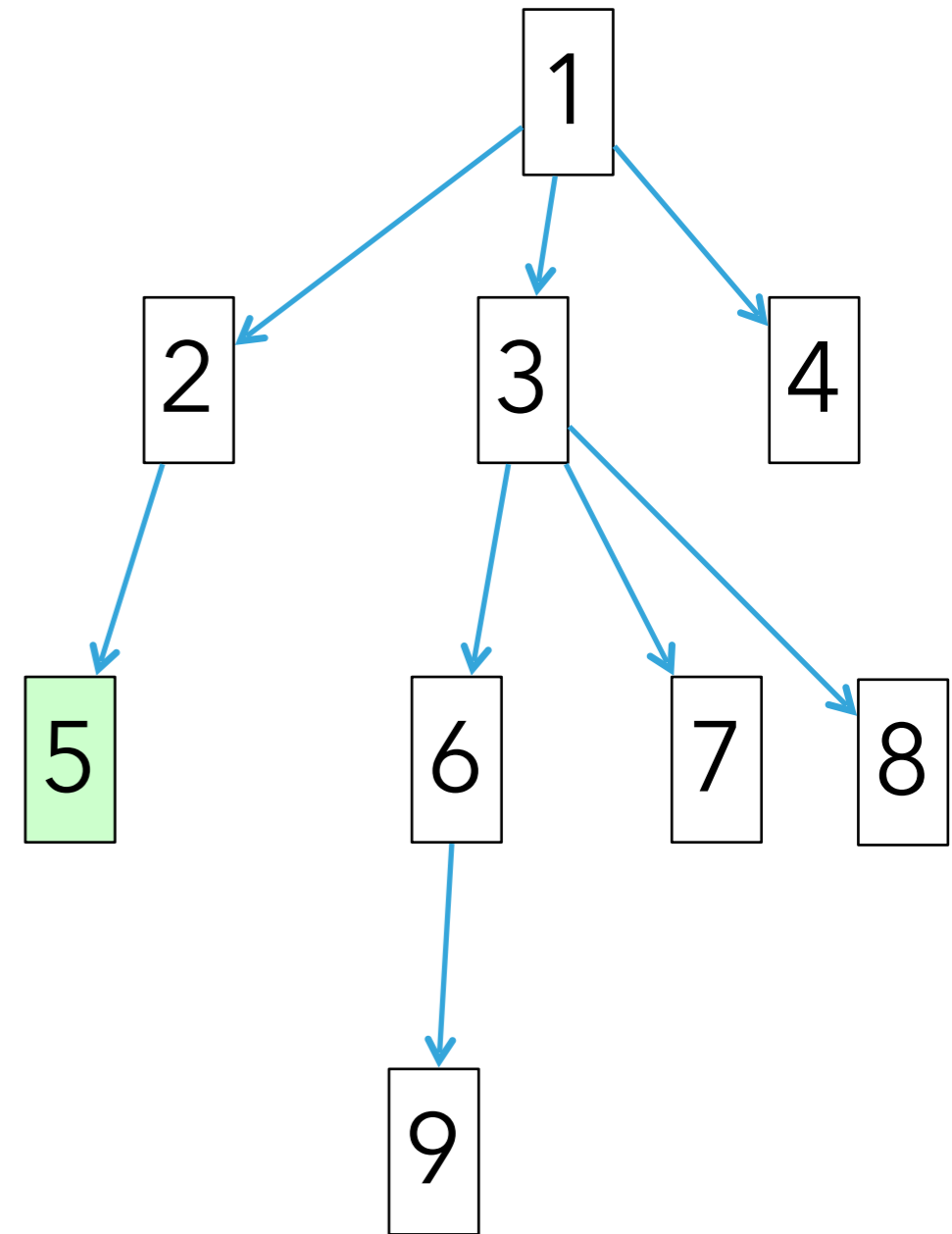
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9



What order will BFS and DFS visit the states?

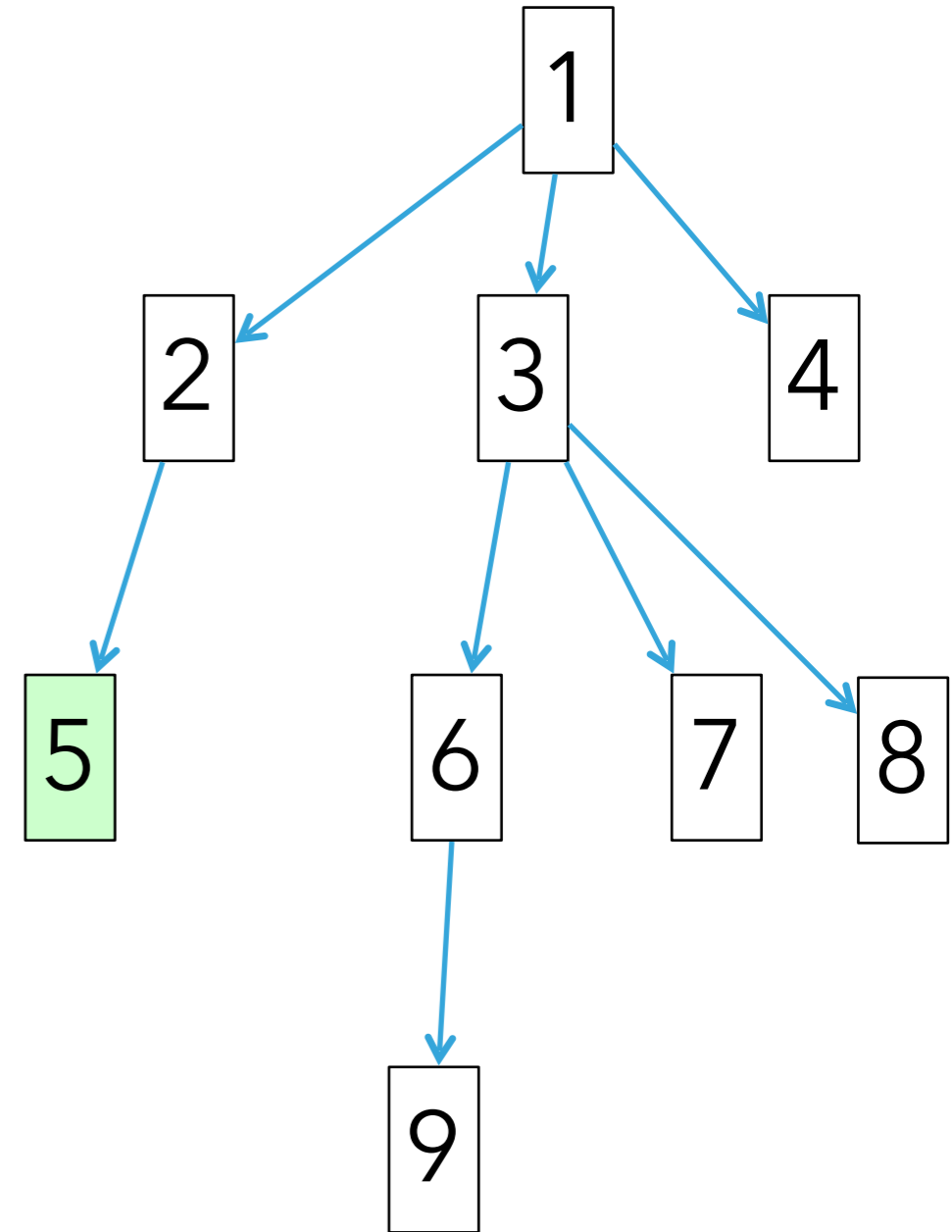
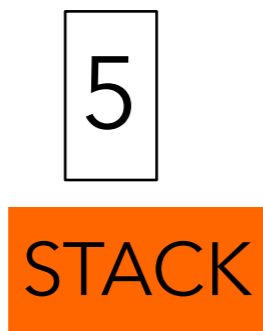
- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2



STACK

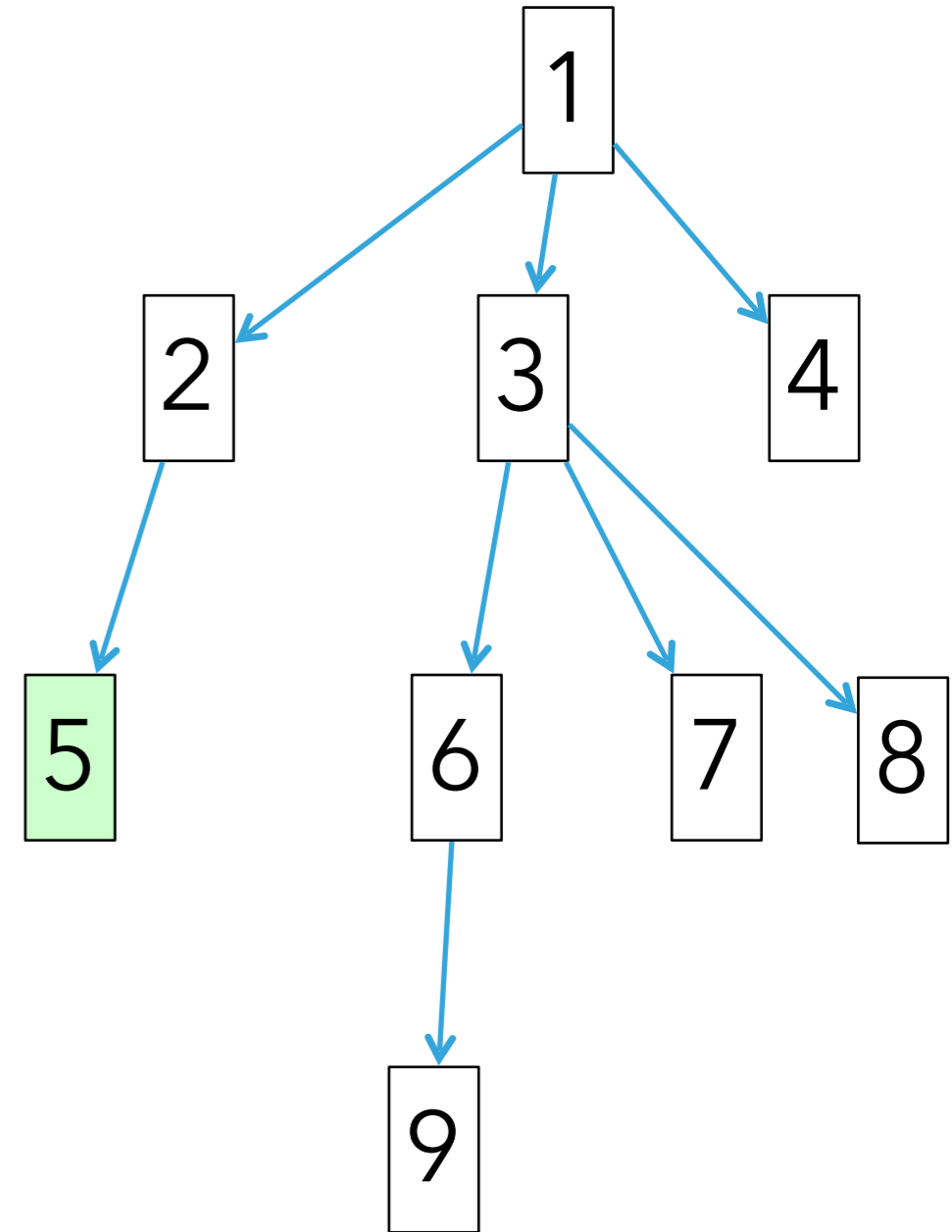
What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2



What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5

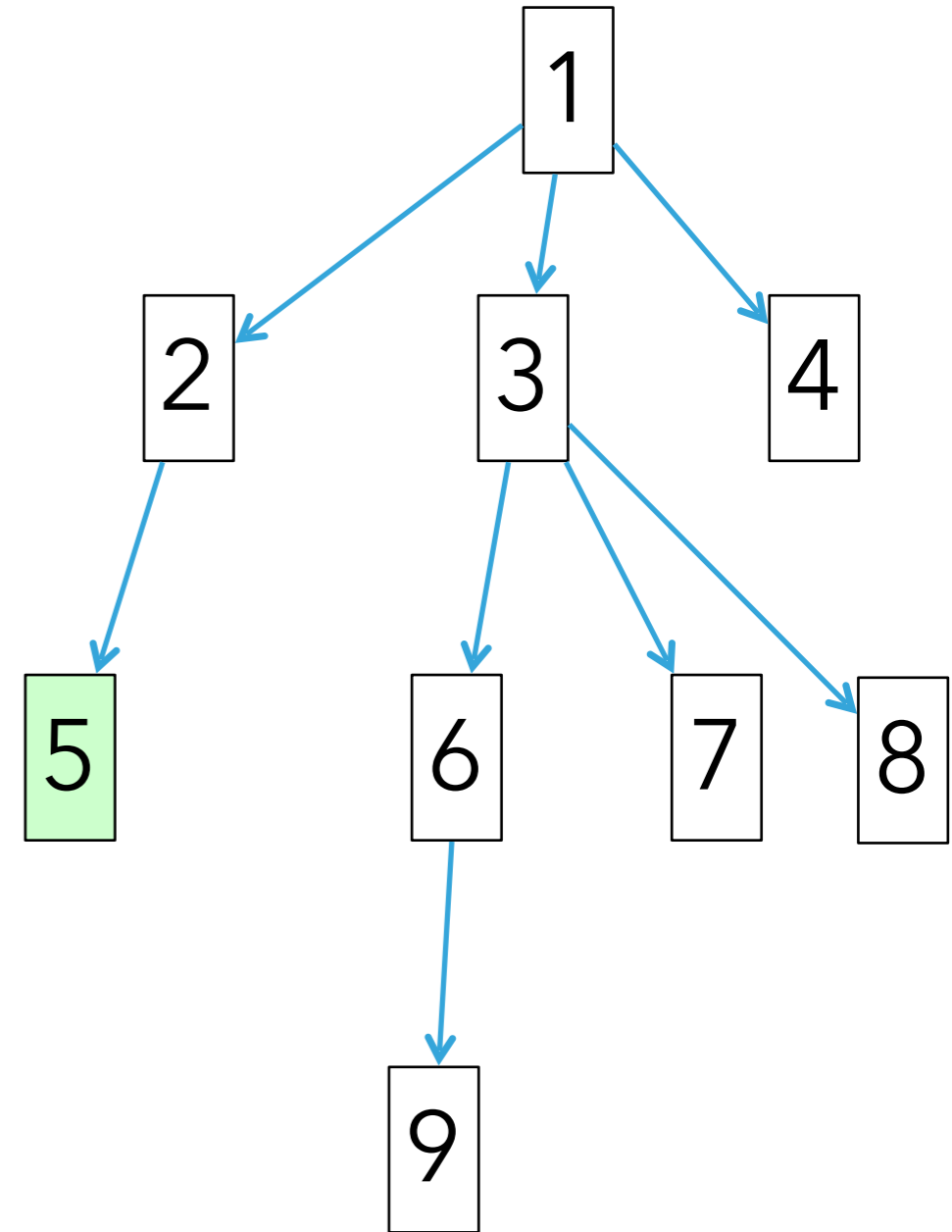


STACK

What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS:

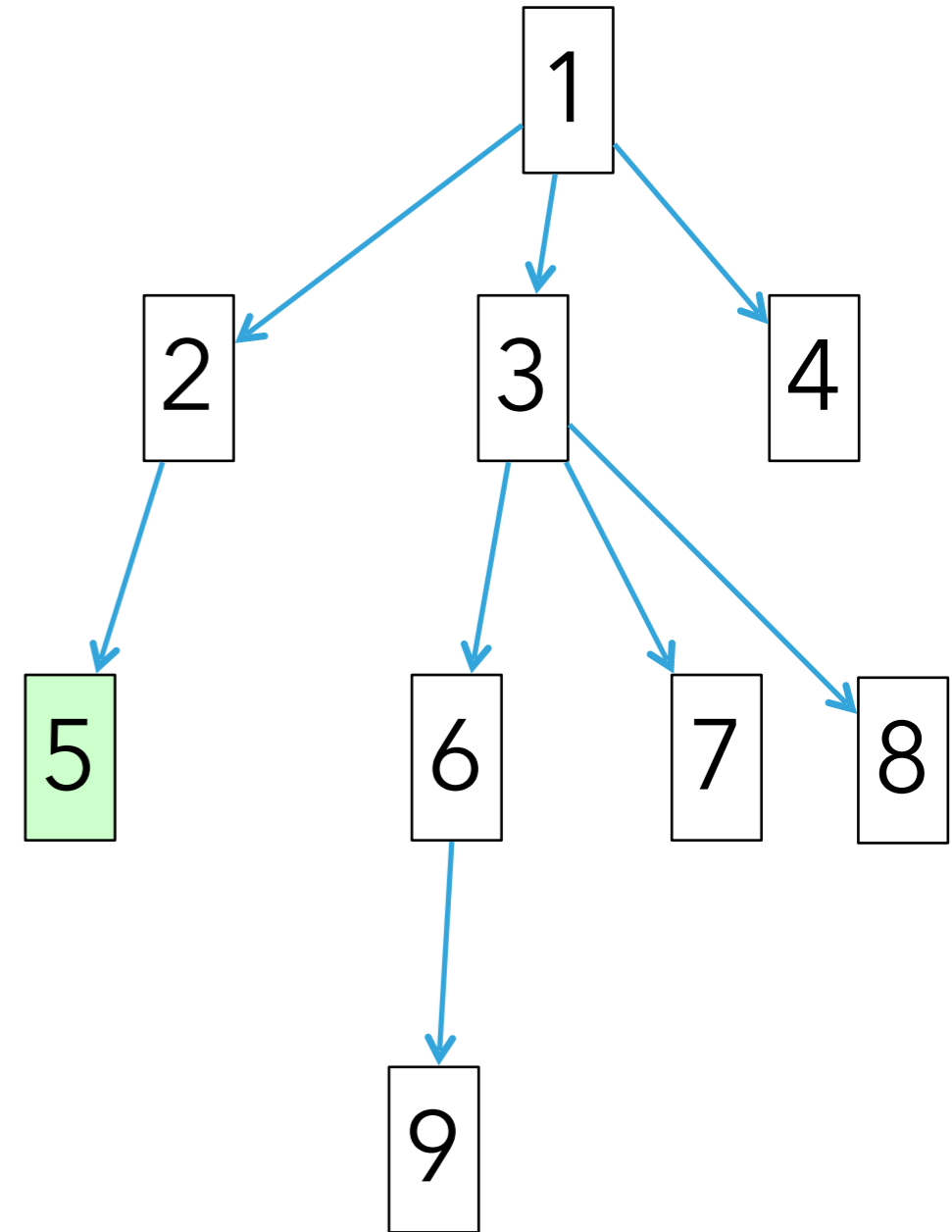
1



QUEUE

What order will BFS and DFS visit the states?

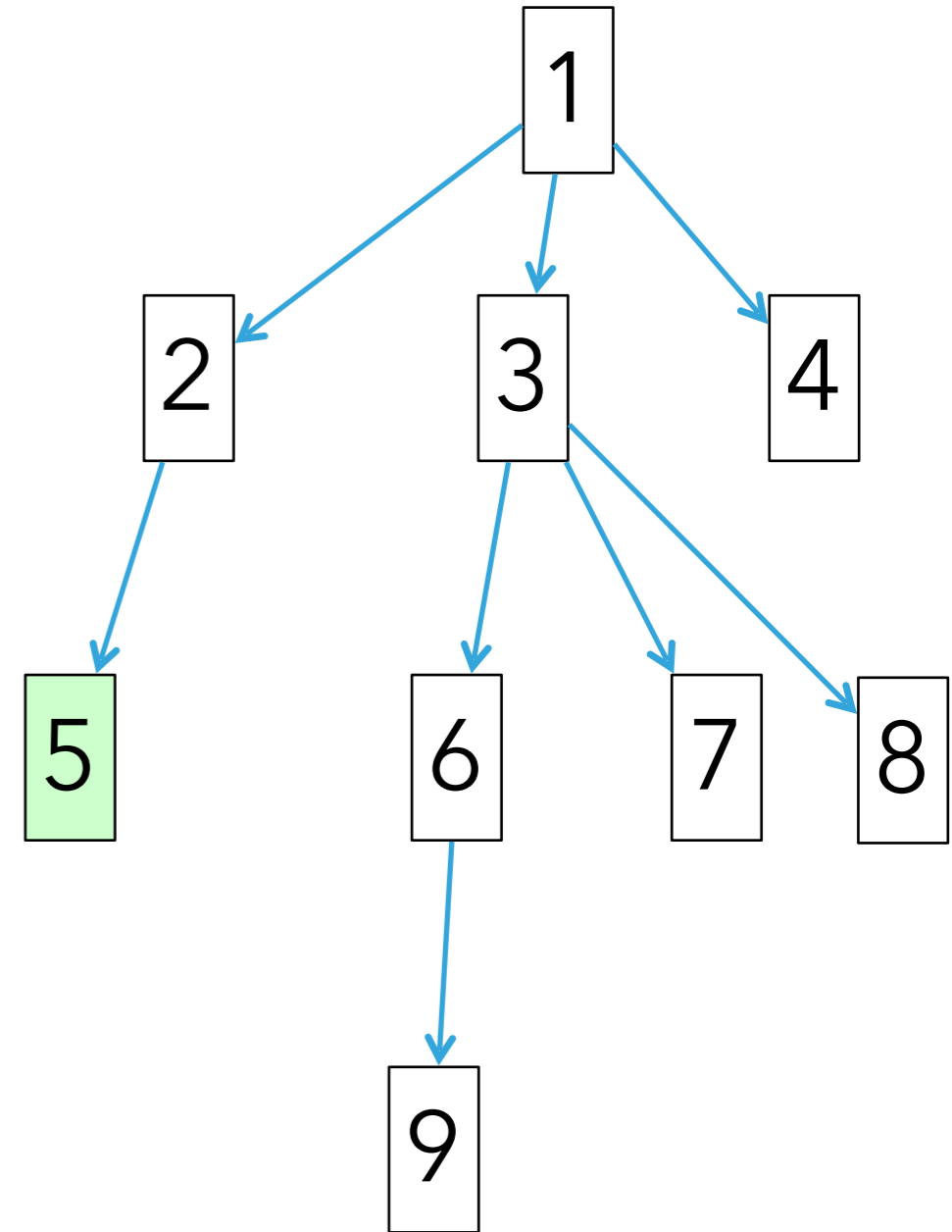
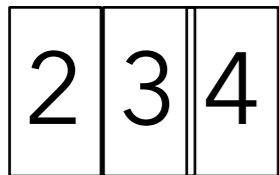
- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1



QUEUE

What order will BFS and DFS visit the states?

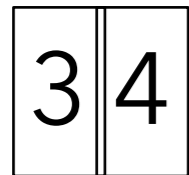
- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1



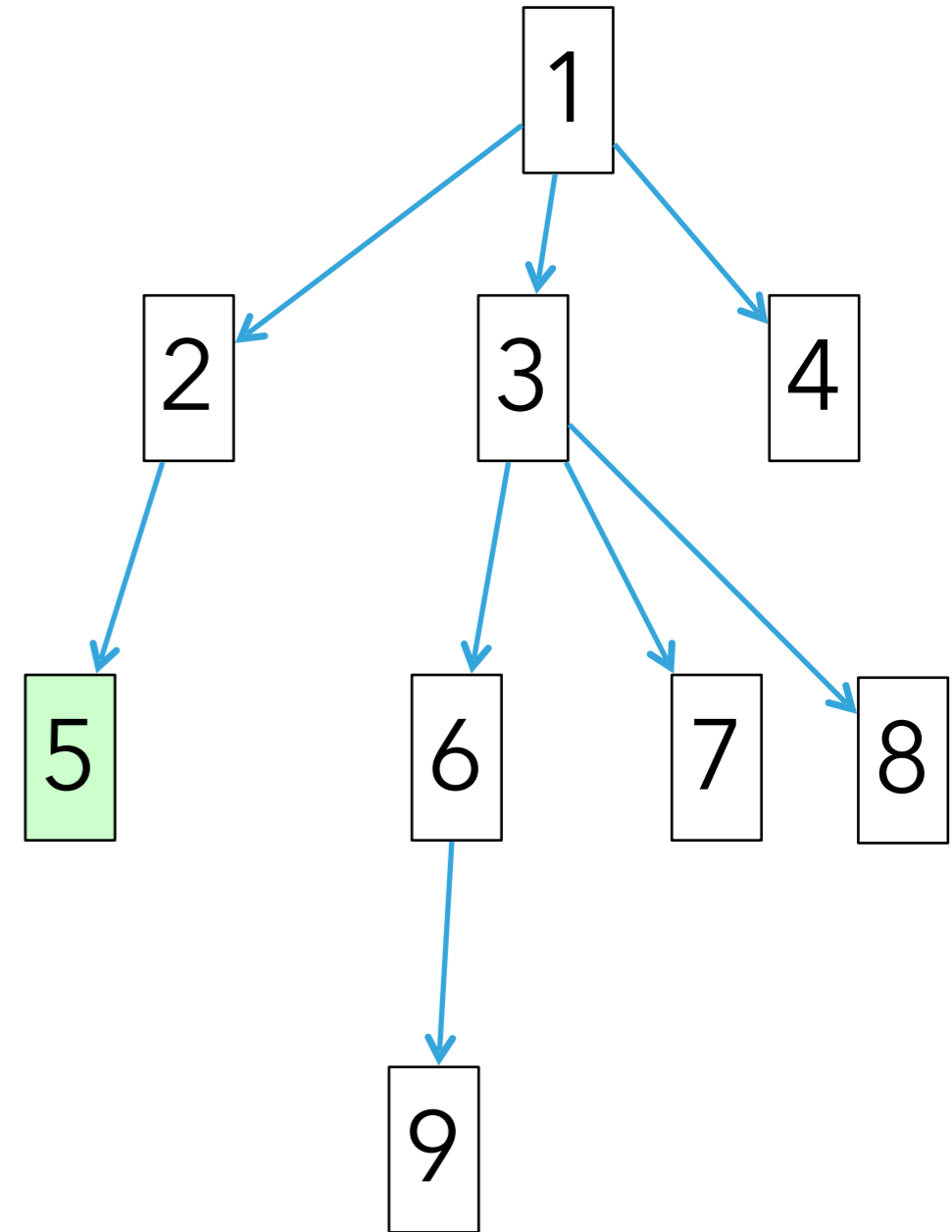
QUEUE

What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1, 2

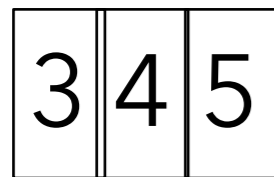


QUEUE

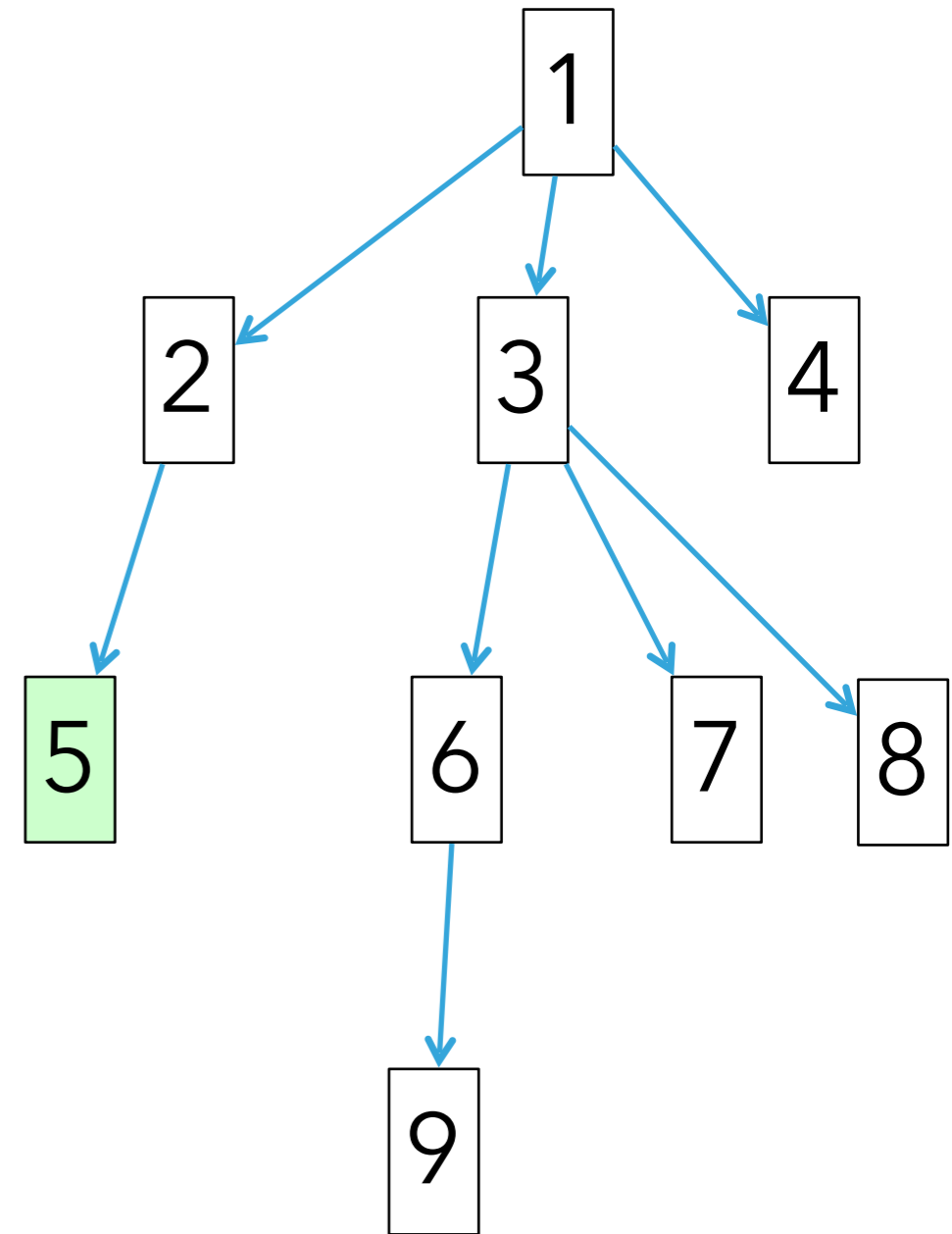


What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1, 2

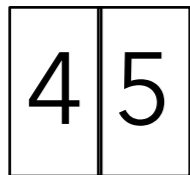


QUEUE

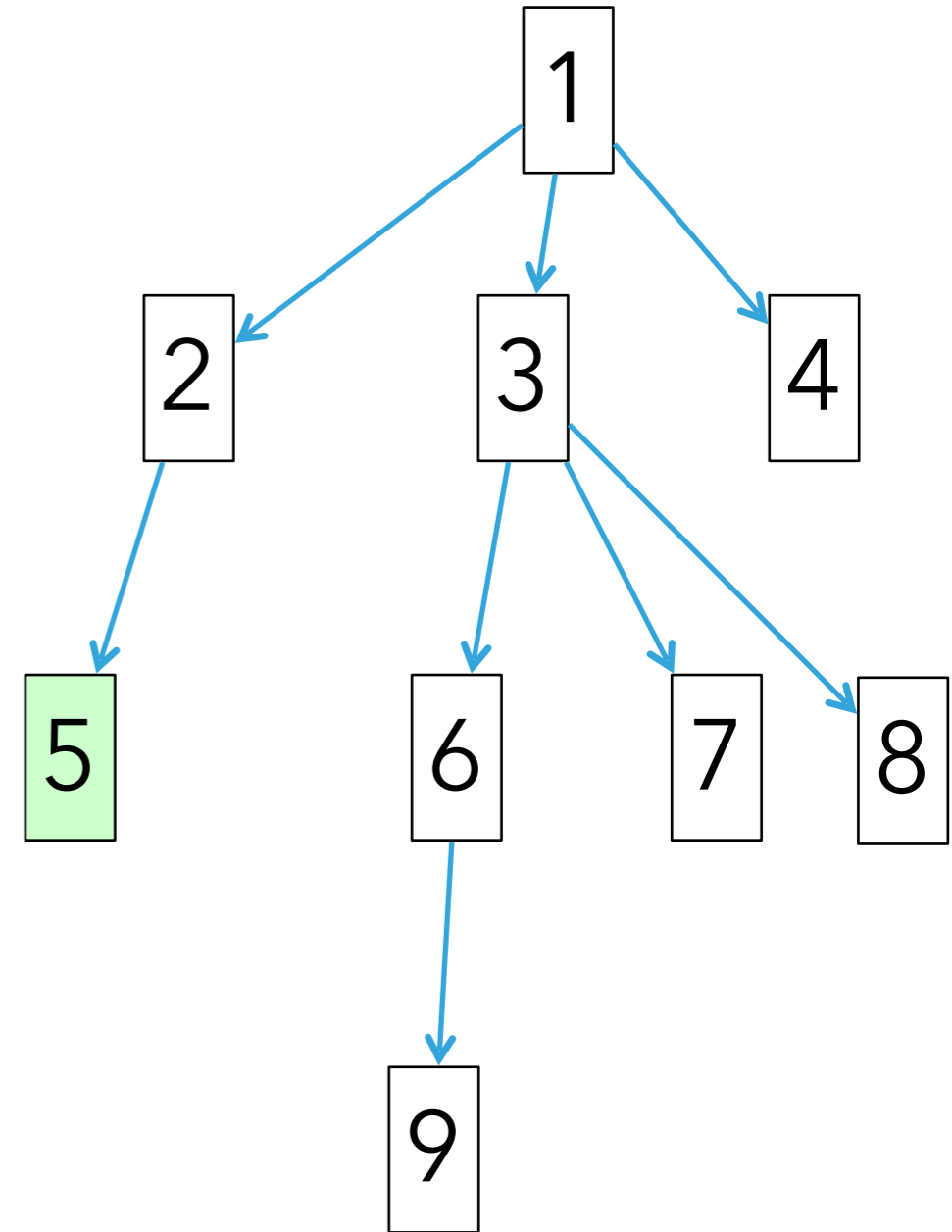


What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1, 2, 3

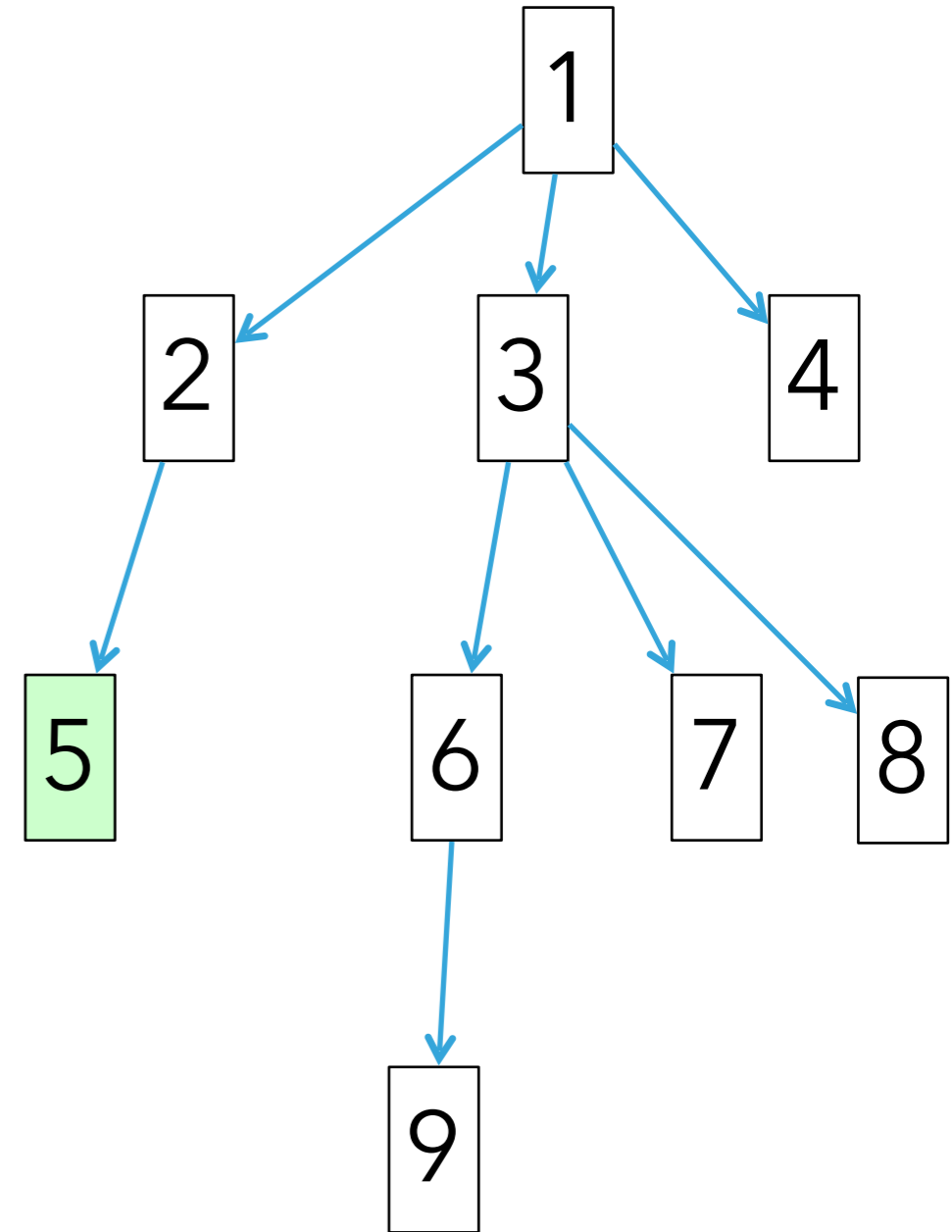
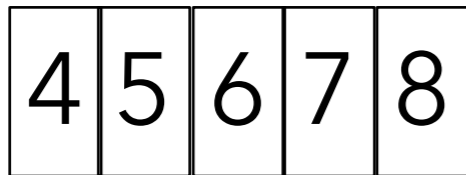


QUEUE



What order will BFS and DFS visit the states?

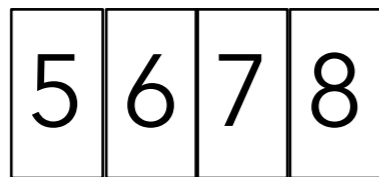
- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1, 2, 3



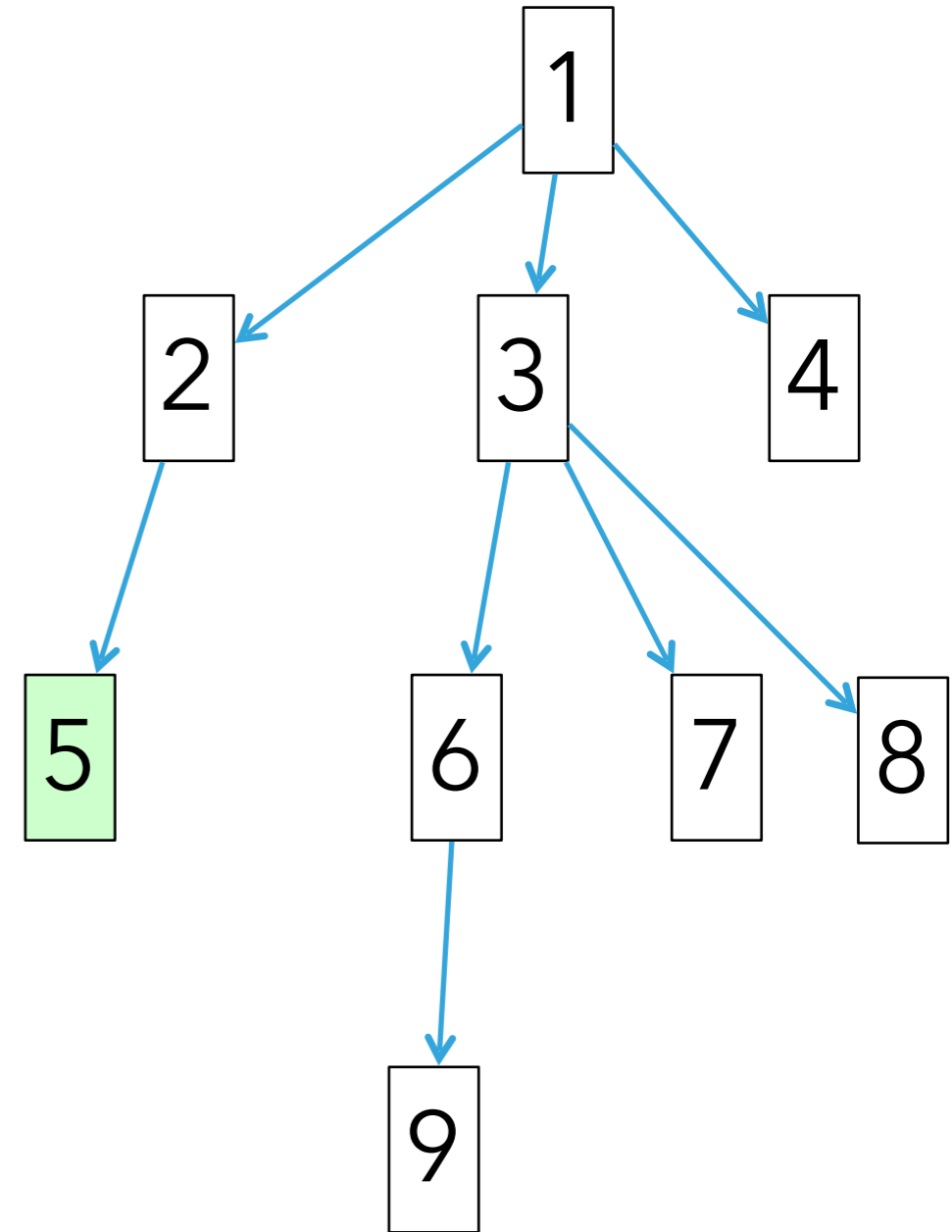
QUEUE

What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1, 2, 3, 4

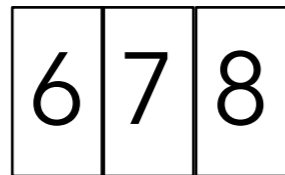


QUEUE

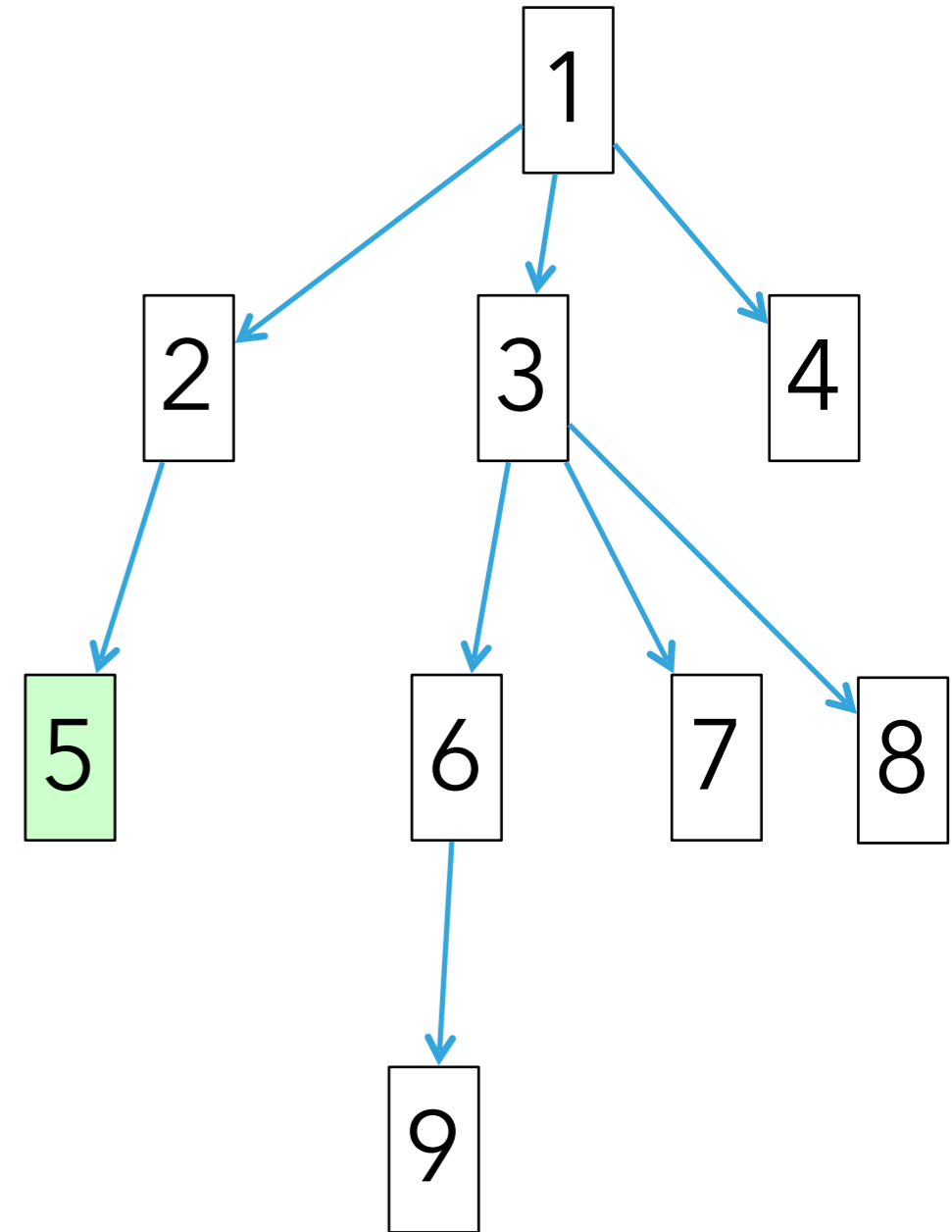


What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1, 2, 3, 4, 5

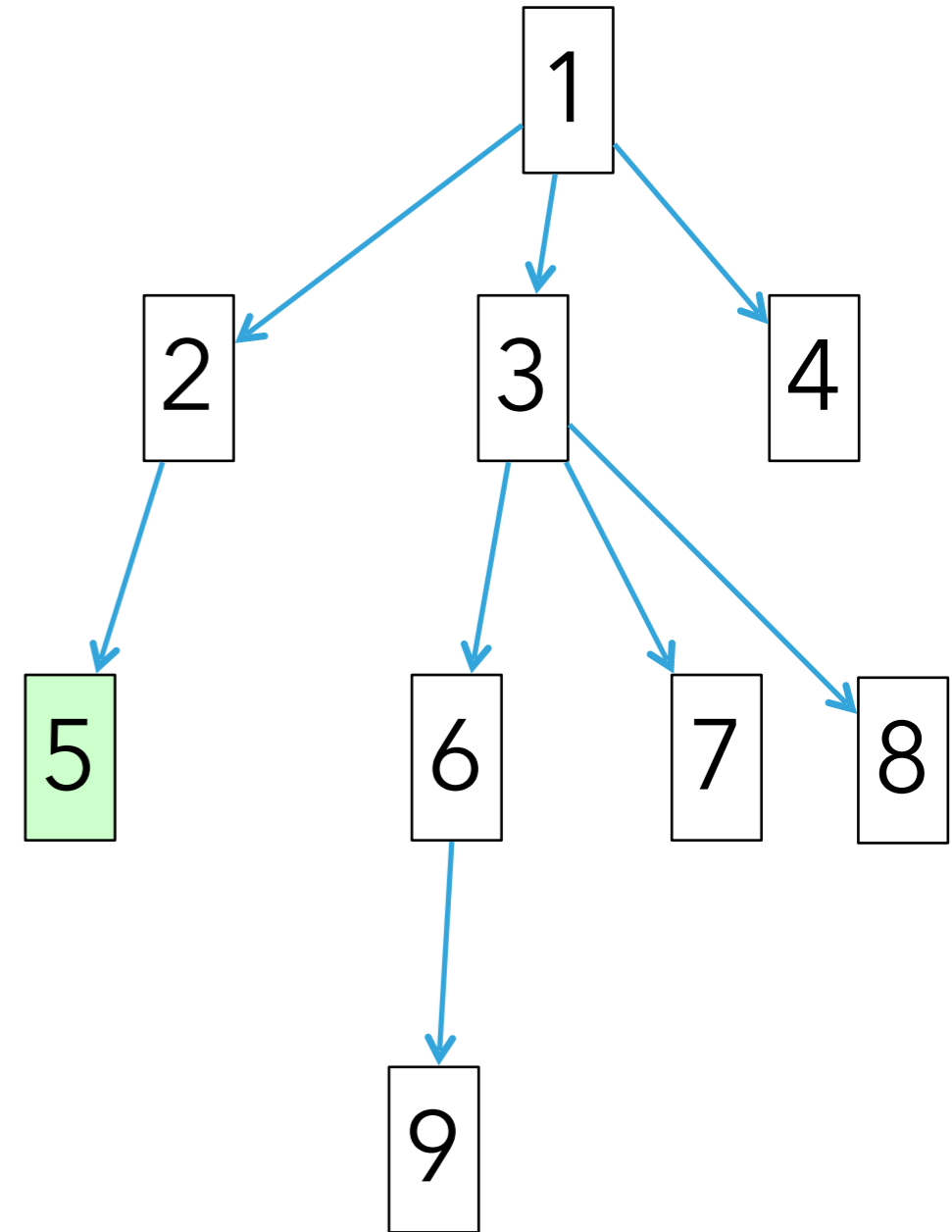


QUEUE



What order will BFS and DFS visit the states?

- ▶ DFS: 1, 4, 3, 8, 7, 6, 9, 2, 5
- ▶ BFS: 1, 2, 3, 4, 5



Homework

- ▶ [Assignment 8](#)