

# 6.009: Fundamentals of Programming

## Week 3 Lecture: Graphs and Graph Search

- Graphs and Graph Search
- BFS and DFS
- Finding Minimum-cost Paths
- Heuristics

Adam Hartz

hz@mit.edu

*9 March 2021*

## Graph Search (Path Finding)

---

- Some set  $V$  of vertices (sometimes called states)
- Some collection  $E$  of edges connecting vertices

## Search Algorithms

---

Example: Eight Puzzle

Start

1	2	3
4	5	6
7	8	

Goal

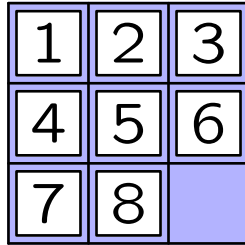
	1	2
3	4	5
6	7	8

Rearrange board by sequentially sliding tiles into the free spot.

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.



## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	3
4	5	
7	8	6

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	3
4		5
7	8	6

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	3
	4	5
7	8	6

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	3
7	4	5
	8	6



## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	3
7	4	5
8		6

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	3
7	4	5
8	6	

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	3
7	4	
8	6	5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	2	
7	4	3
8	6	5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1		2
7	4	3
8	6	5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
7		3
8	6	5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
7	6	3
8		5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
7	6	3
	8	5



## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
	6	3
7	8	5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
6		3
7	8	5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
6	3	
7	8	5

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
6	3	5
7	8	

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
6	3	5
7		8

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
6	3	5
	7	8

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
	3	5
6	7	8

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1	4	2
3		5
6	7	8



## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

1		2
3	4	5
6	7	8

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

	1	2
3	4	5
6	7	8

## Eight Puzzle

---

Rearrange board by sequentially sliding tiles into the free spot.

	1	2
3	4	5
6	7	8

That solution took twenty-two moves.

How difficult is this problem?

## Check Yourself!

---

1	2	3
4	5	6
7	8	

	1	2
3	4	5
6	7	8

How many different board configurations (vertices) exist?

1.  $8^2 = 64$
2.  $9^2 = 81$
3.  $8! = 40,320$
4.  $9! = 362,880$

## Eight Puzzle

---

1	2	3
4	5	6
7	8	

	1	2
3	4	5
6	7	8

Is the solution with 22 moves optimal? Do shorter solutions exist?

Do we have to look at all 362,880 configurations to be sure?

## Search Algorithm

---

Develop an algorithm to systematically conduct a search.

Analyze how well the algorithm performs.

Optimize the algorithm:

- find the “best” solution (i.e., minimum path length)
- by considering as few cases as possible.

## Pathfinding Algorithm

---

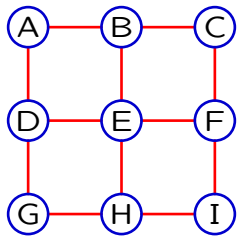
Pathfinding Algorithm:

- Initialize **visited set** (all vertices that have been added to the agenda)
  - Initialize **agenda** (list of paths to consider)
  - Repeat the following:
    - Remove one path from the agenda
    - For each child (of that path's terminal vertex):
      - ★ If it satisfies the goal condition, return a path.
      - ★ Otherwise, if it is not already in the visited set:
        - ▷ Add the new path to the agenda
        - ▷ Add the new vertex to the visited set
- until **goal is found** or **agenda is empty**

## Example

---

Find shortest path between 2 points on a rectangular grid.

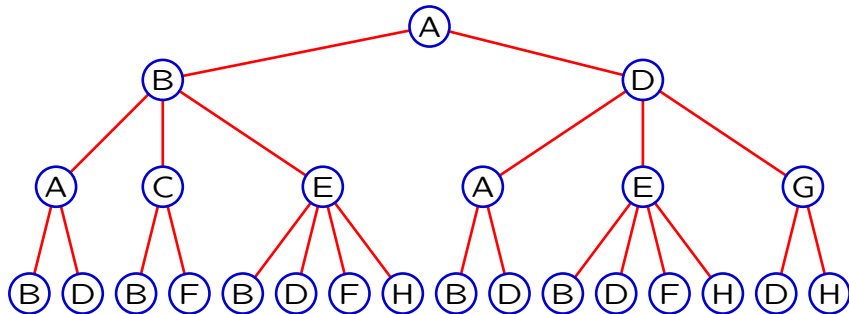




## Order Matters

---

Replace first path in agenda by its children:

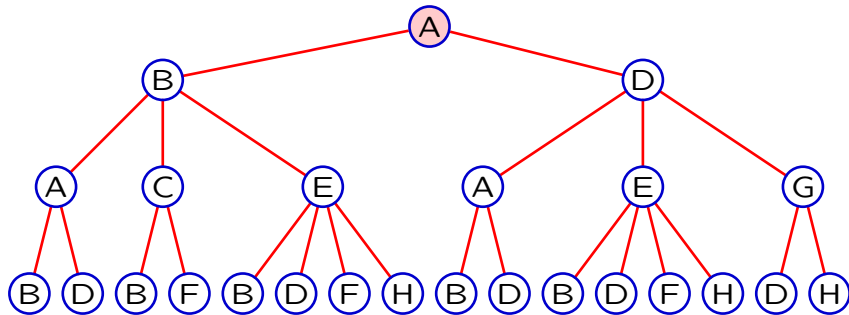


step    Agenda

## Order Matters

---

Replace first path in agenda by its children:



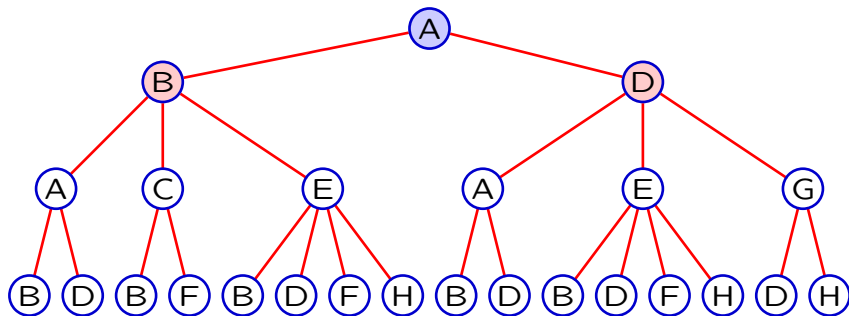
step    Agenda

0:     **A**

## Order Matters

---

Replace first path in agenda by its children:



step    Agenda

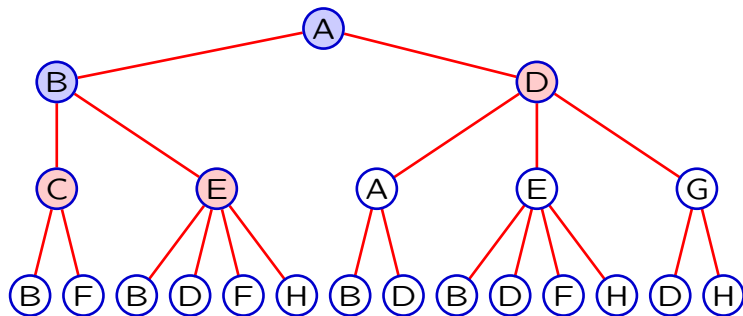
0:     A

1:     AB AD

## Order Matters

---

Replace first path in agenda by its children:



step    Agenda

0:     A

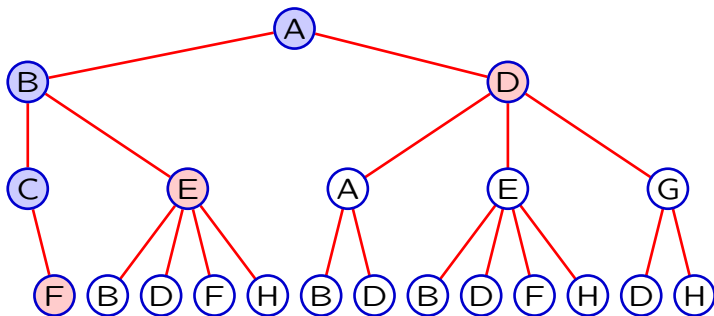
1:     AB AD

2:     ABC ABE AD

## Order Matters

---

Replace first path in agenda by its children:



step    Agenda

0:     A

1:     AB AD

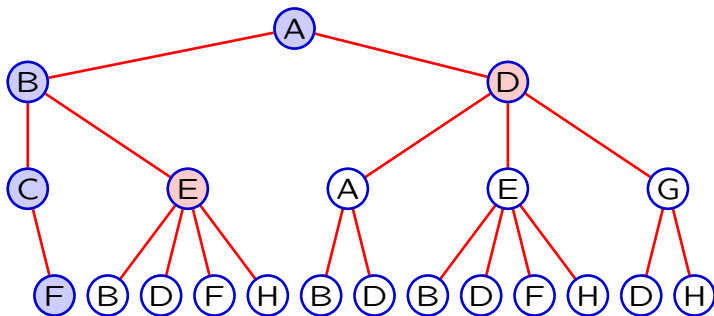
2:     ABC ABE AD

3:     ABCF ABE AD

## Order Matters

---

Replace first path in agenda by its children:

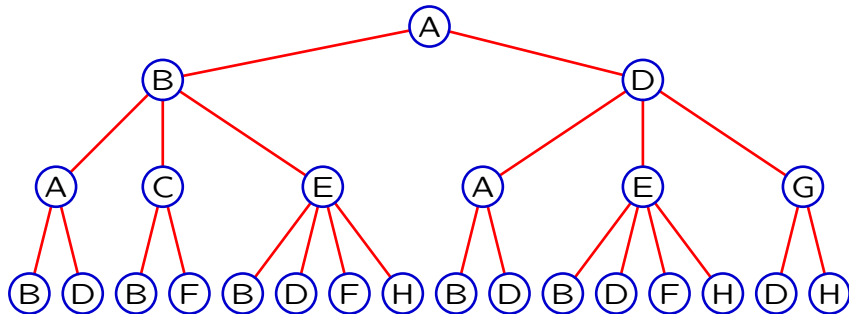


**Depth-first Search**

## Order Matters

---

Replace last path in agenda by its children:

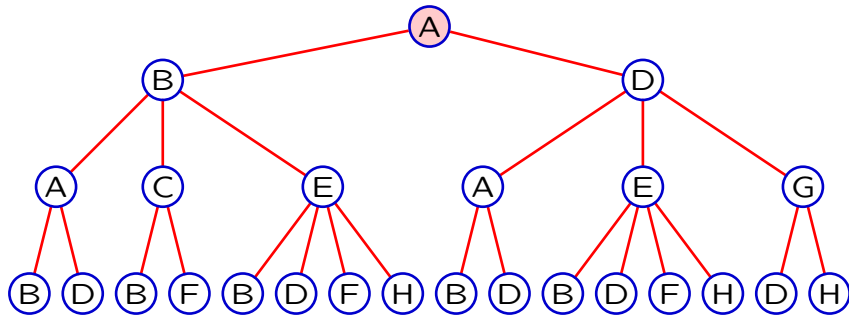


step   Agenda

## Order Matters

---

Replace last path in agenda by its children:



step    Agenda

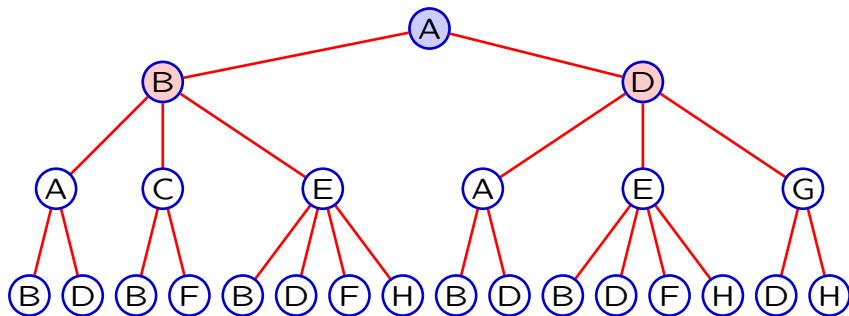
0:     **A**



## Order Matters

---

Replace last path in agenda by its children:



step    Agenda

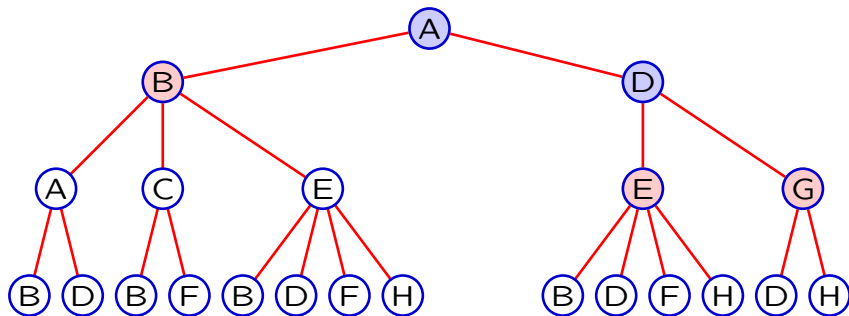
0:     **A**

1:     **AB AD**

## Order Matters

---

Replace last path in agenda by its children:



step    Agenda

0:     A

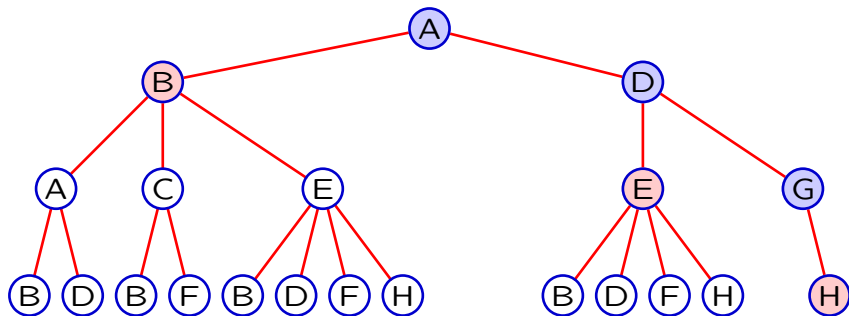
1:     AB AD

2:     AB ADE ADG

## Order Matters

---

Replace last path in agenda by its children:



step    Agenda

0:     A

1:     AB AD

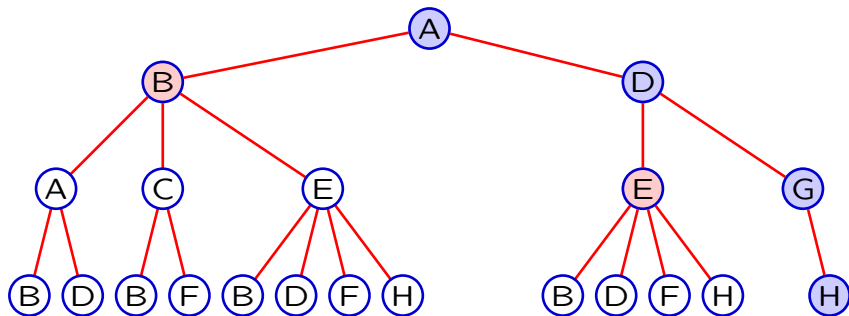
2:     AB ADE ADG

3:     AB ADE ADGH

## Order Matters

---

Replace last path in agenda by its children:

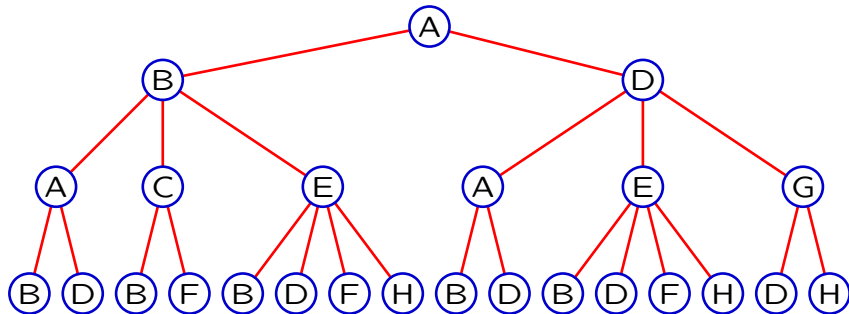


also **Depth-first Search**

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.

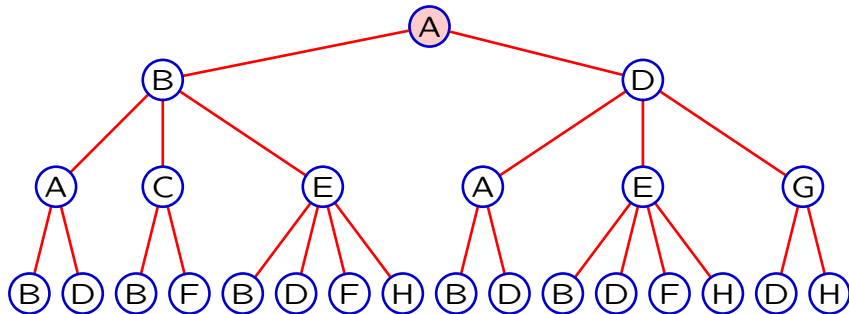


step   Agenda

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.



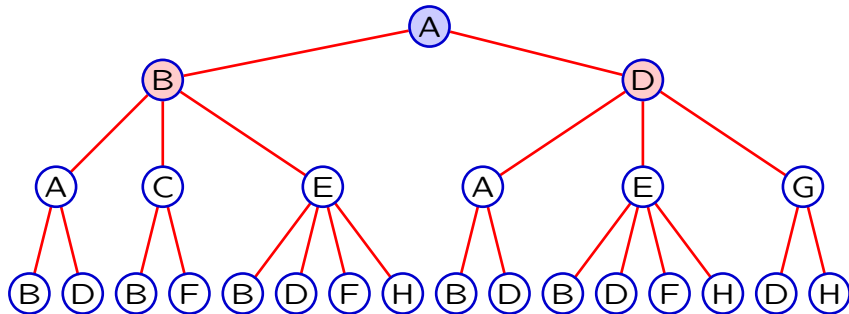
step    Agenda

0:     **A**

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.



step    Agenda

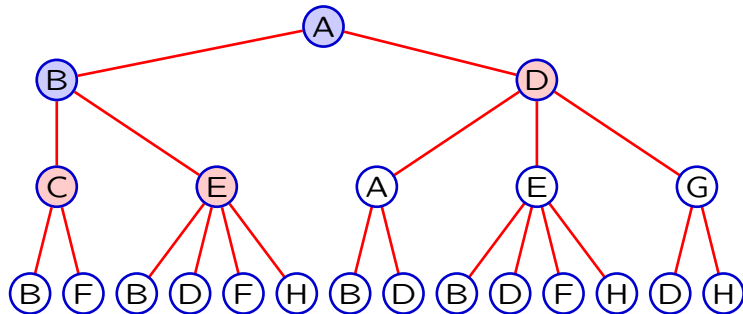
0:     A

1:     AB AD

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.



step    Agenda

0:      A

1:      AB AD

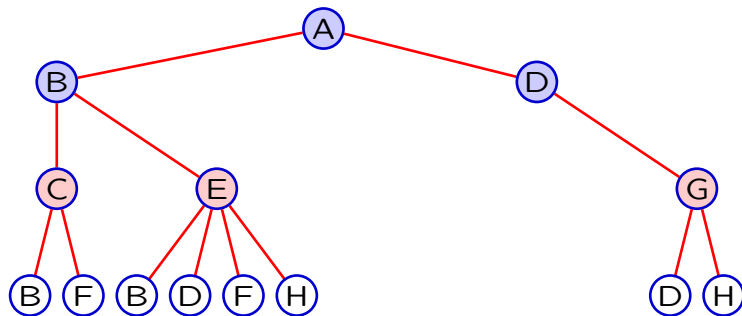
2:      AD ABC ABE



## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.

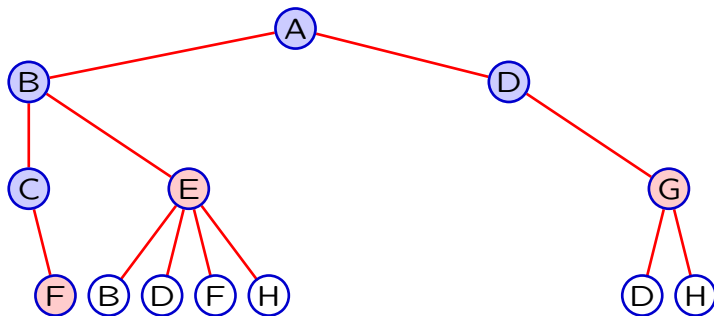


step	Agenda
0:	A
1:	AB AD
2:	AD ABC ABE
3:	ABC ABE ADG

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.



step    Agenda

1:      AB AD

2:      AD ABC ABE

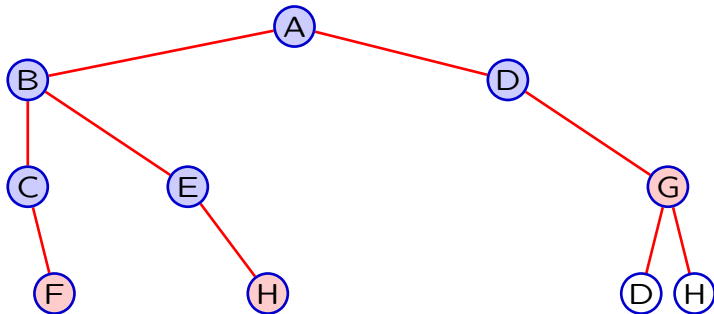
3:      **ABC** ABE ADG

4:      ABE ADG **ABCF**

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.

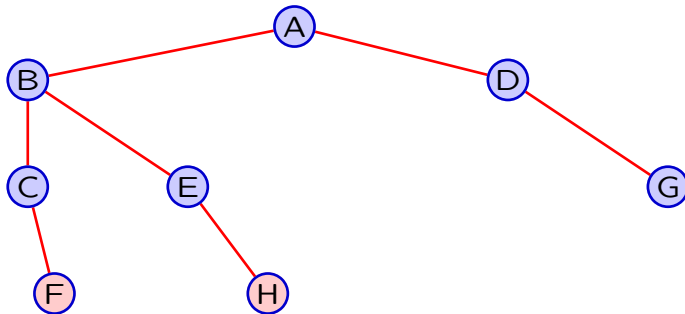


step	Agenda
2:	AD ABC ABE
3:	ABC ABE ADG
4:	ABE ADG ABCF
5:	ADG ABCF ABEH

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.

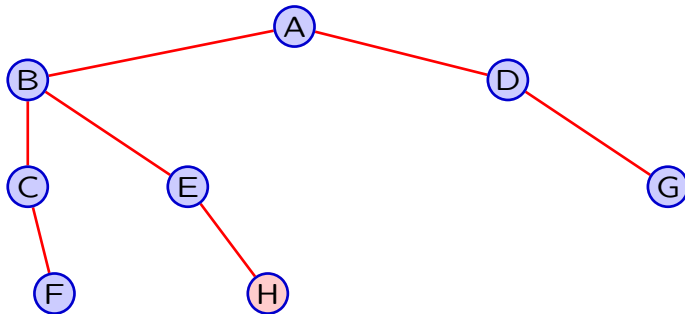


step	Agenda
3:	ABC ABE ADG
4:	ABE ADG ABCF
5:	ADG ABCF ABEH
6:	ABCF ABEH

## Order Matters

---

Remove first path from agenda. Add its children to end of agenda.

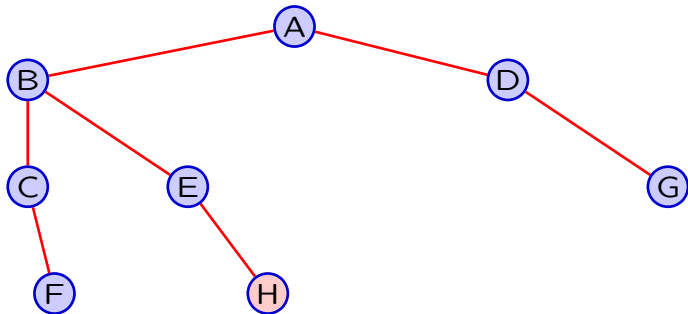


step	Agenda
4:	ABE ADG ABCF
5:	ADG ABCF ABEH
6:	ABCF ABEH
7:	ABEH ABCFI

## Order Matters

---

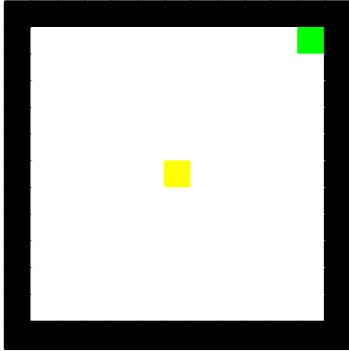
Remove first path from agenda. Add its children to end of agenda.



**Breadth-first Search**

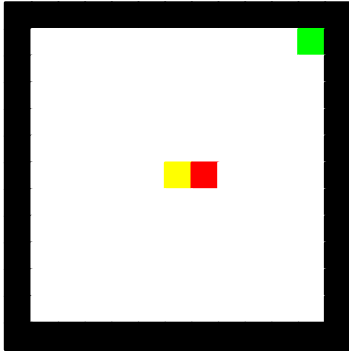
## Grid: BFS 1

---



## Grid: BFS 1

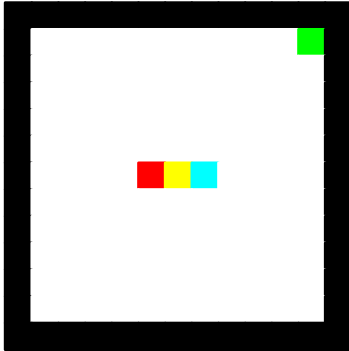
---





## Grid: BFS 1

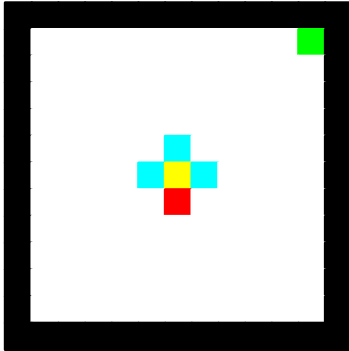
---



**Grid: BFS 1**

## Grid: BFS 1

---

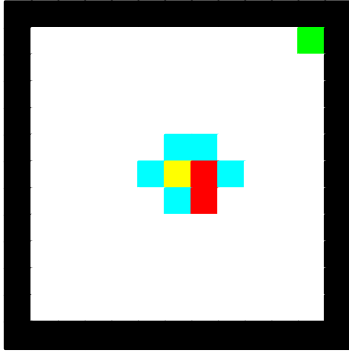


**Grid: BFS 1**

**Grid: BFS 1**

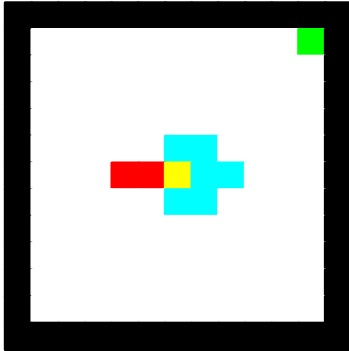
## Grid: BFS 1

---



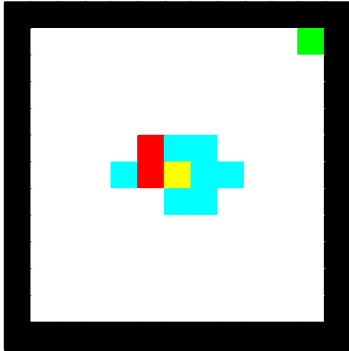
## Grid: BFS 1

---



## Grid: BFS 1

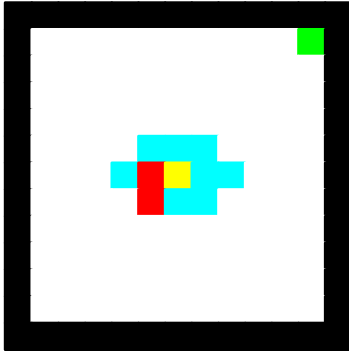
---





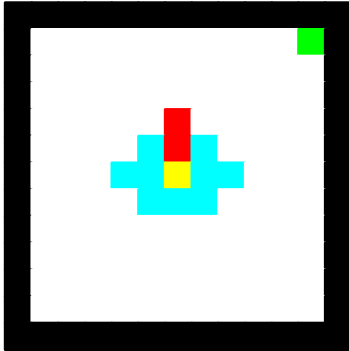
## Grid: BFS 1

---



## Grid: BFS 1

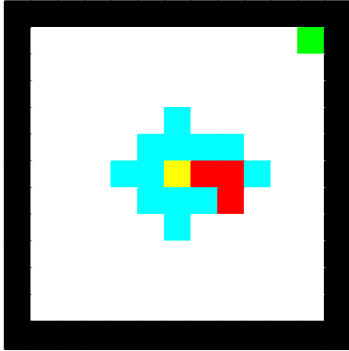
---



**Grid: BFS 1**

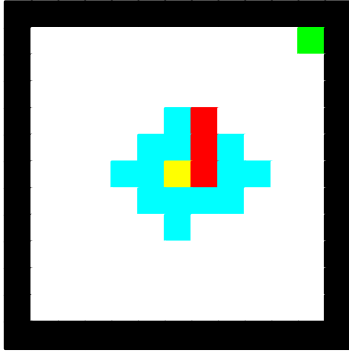
**Grid: BFS 1**

**Grid: BFS 1**



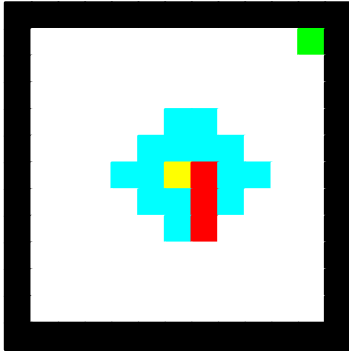
## Grid: BFS 1

---



## Grid: BFS 1

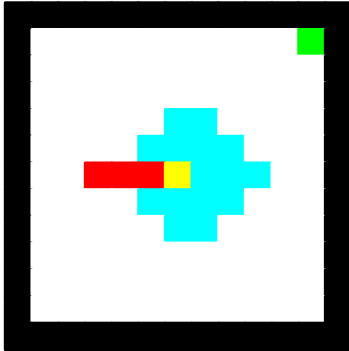
---



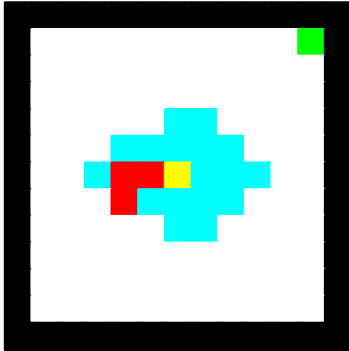


## Grid: BFS 1

---

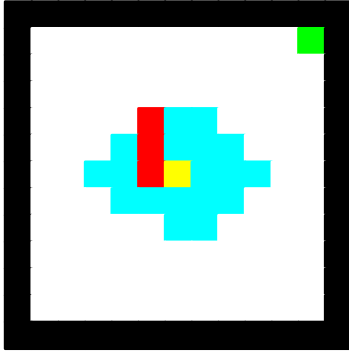


**Grid: BFS 1**



## Grid: BFS 1

---

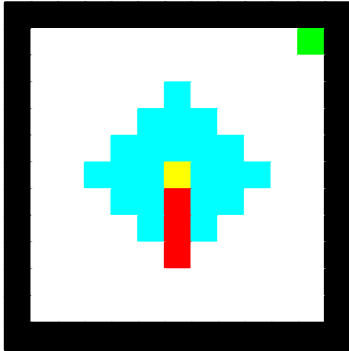


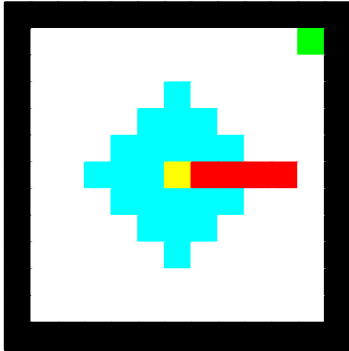
**Grid: BFS 1**

**Grid: BFS 1**

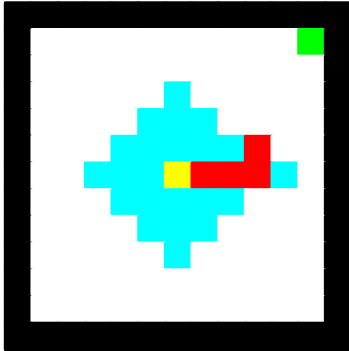
## Grid: BFS 1

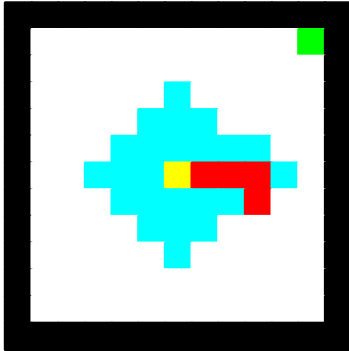
---





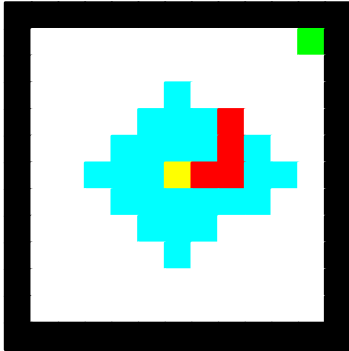






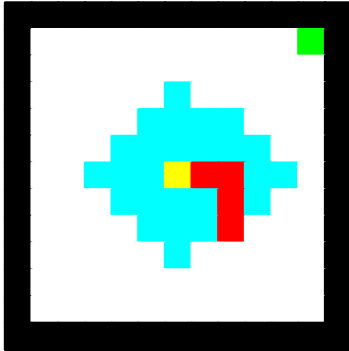
## Grid: BFS 1

---



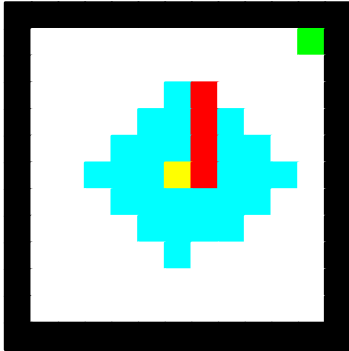
## Grid: BFS 1

---



## Grid: BFS 1

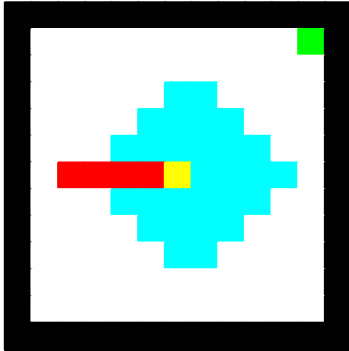
---



**Grid: BFS 1**

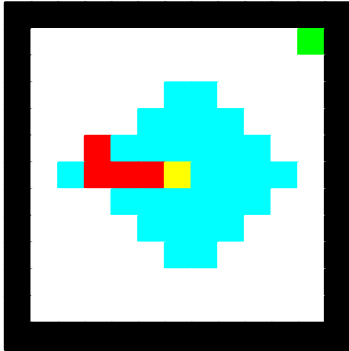
## Grid: BFS 1

---



## Grid: BFS 1

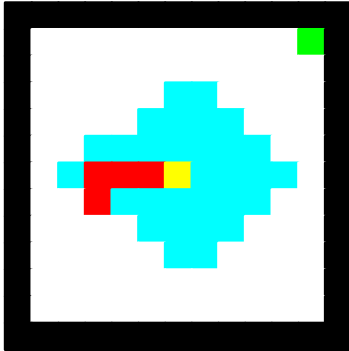
---





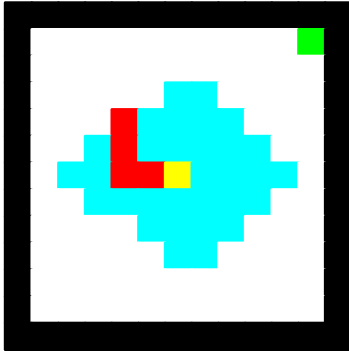
## Grid: BFS 1

---



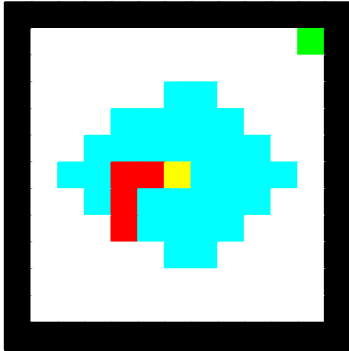
## Grid: BFS 1

---



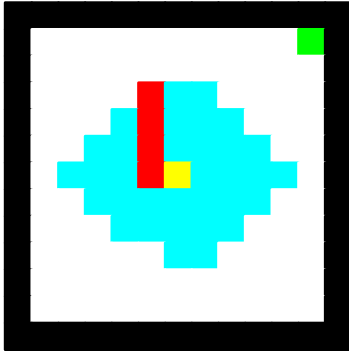
## Grid: BFS 1

---



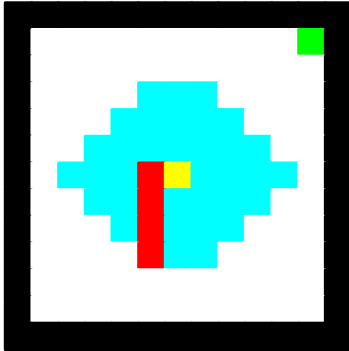
## Grid: BFS 1

---



## Grid: BFS 1

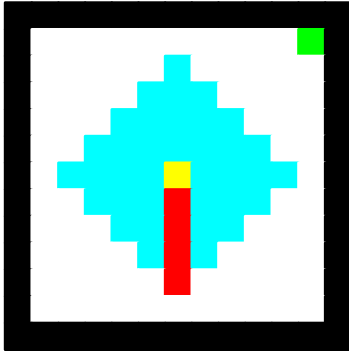
---

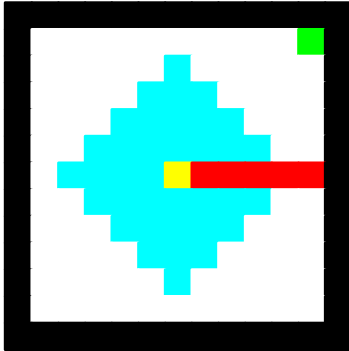


**Grid: BFS 1**

## Grid: BFS 1

---

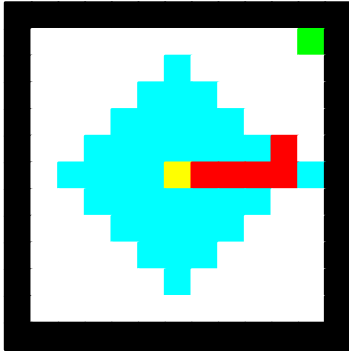


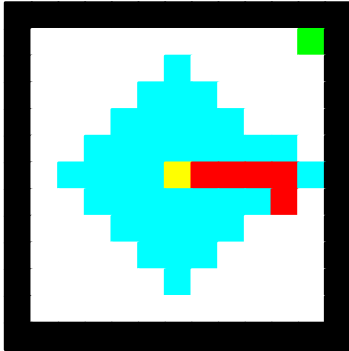




## Grid: BFS 1

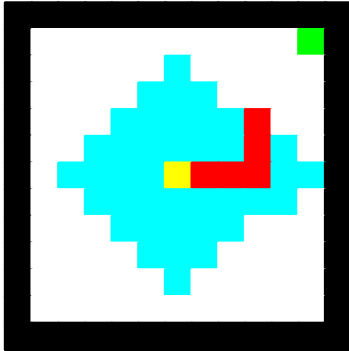
---





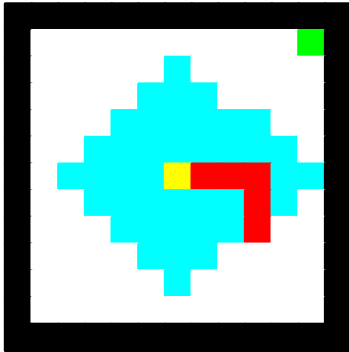
## Grid: BFS 1

---



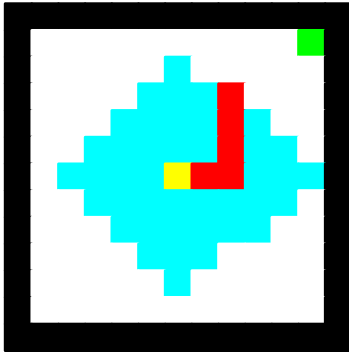
## Grid: BFS 1

---



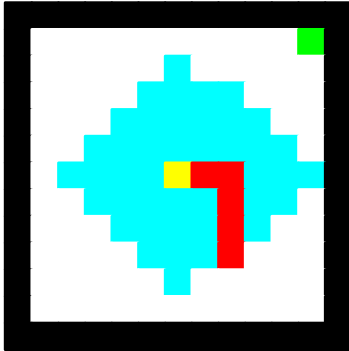
## Grid: BFS 1

---



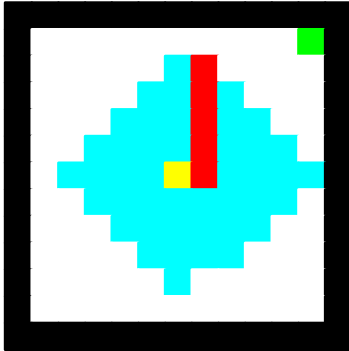
## Grid: BFS 1

---



## Grid: BFS 1

---

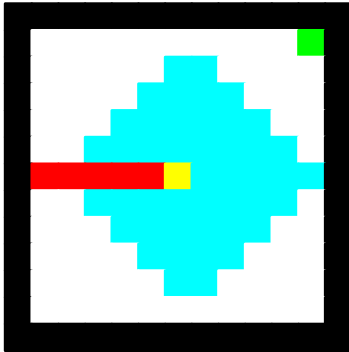


**Grid: BFS 1**



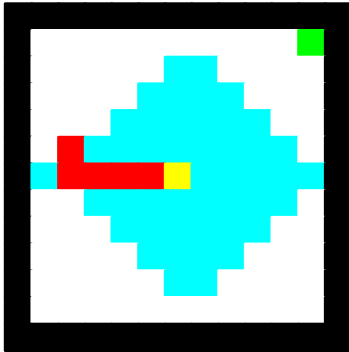
## Grid: BFS 1

---



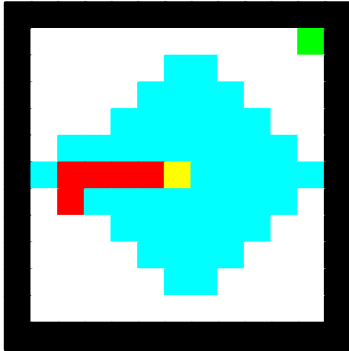
## Grid: BFS 1

---



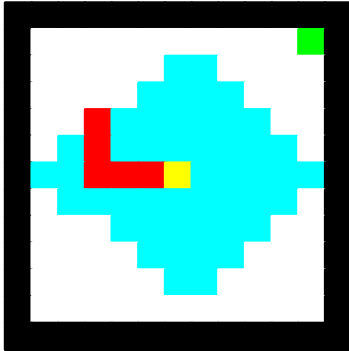
## Grid: BFS 1

---



## Grid: BFS 1

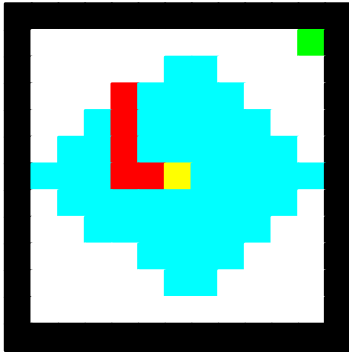
---



**Grid: BFS 1**

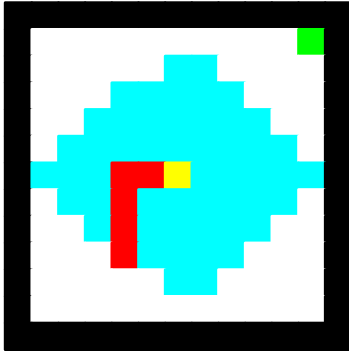
## Grid: BFS 1

---



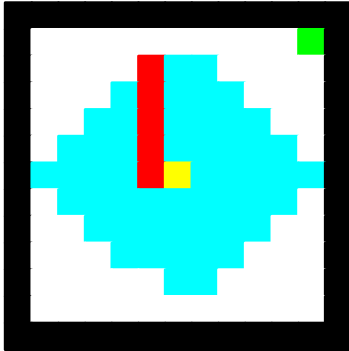
## Grid: BFS 1

---



## Grid: BFS 1

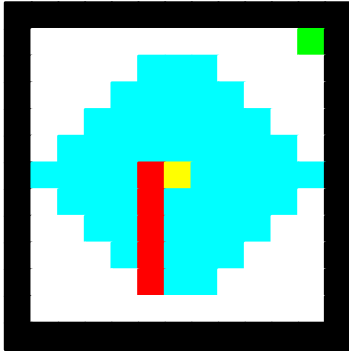
---





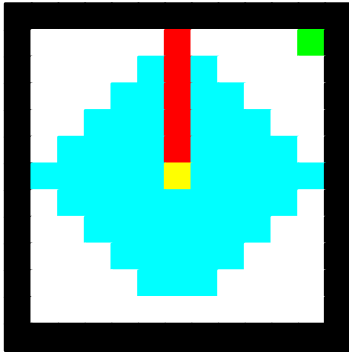
## Grid: BFS 1

---



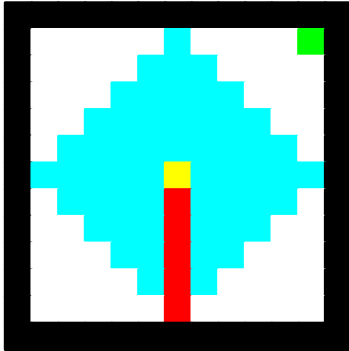
## Grid: BFS 1

---

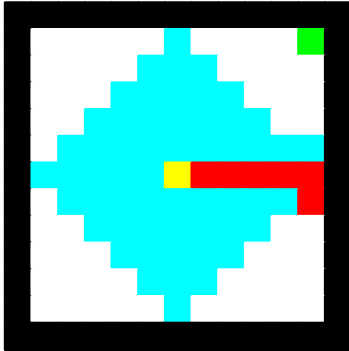


## Grid: BFS 1

---



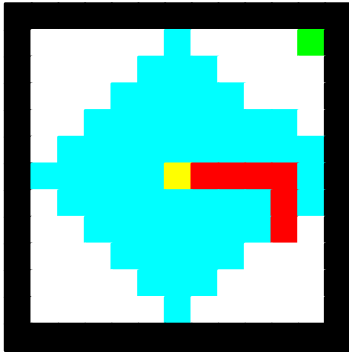
**Grid: BFS 1**



**Grid: BFS 1**

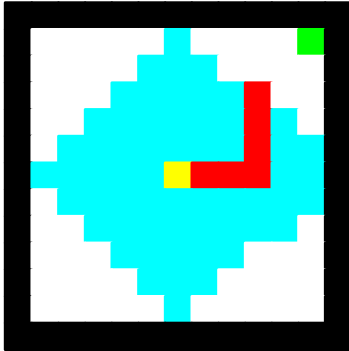
## Grid: BFS 1

---



## Grid: BFS 1

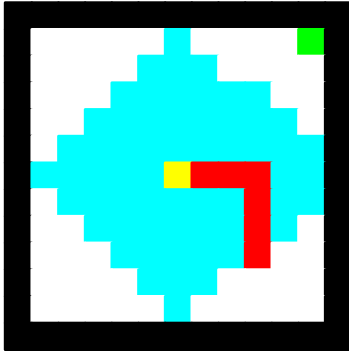
---





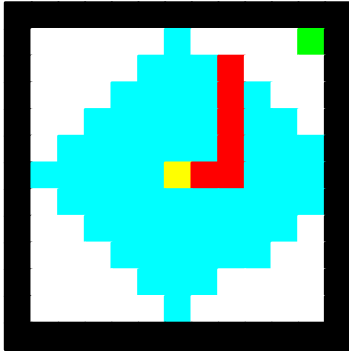
## Grid: BFS 1

---



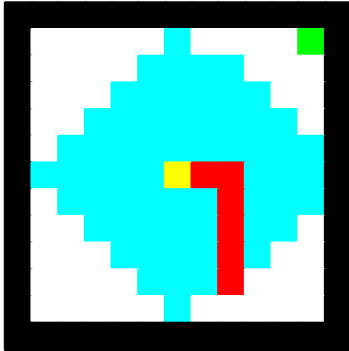
## Grid: BFS 1

---



## Grid: BFS 1

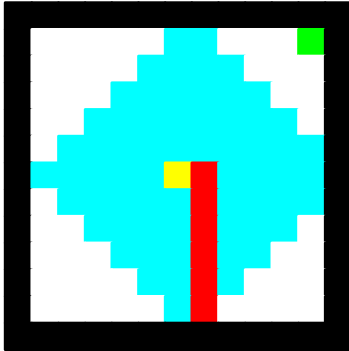
---



**Grid: BFS 1**

## Grid: BFS 1

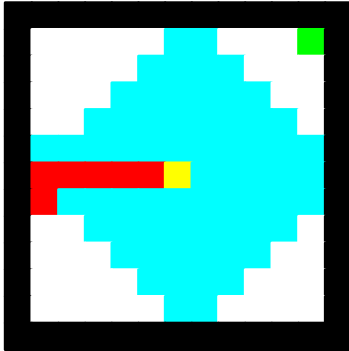
---



**Grid: BFS 1**

## Grid: BFS 1

---

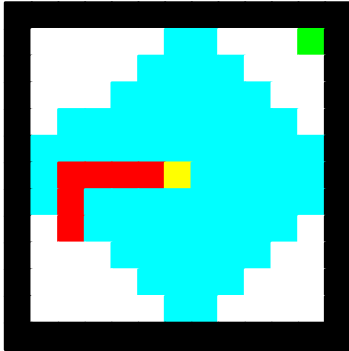


**Grid: BFS 1**



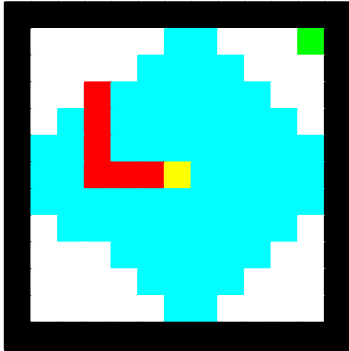
## Grid: BFS 1

---



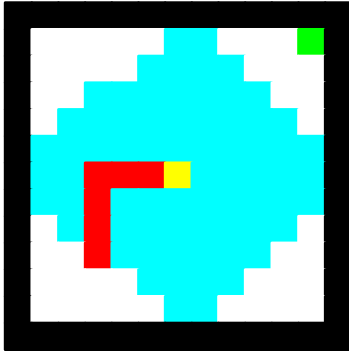
## Grid: BFS 1

---



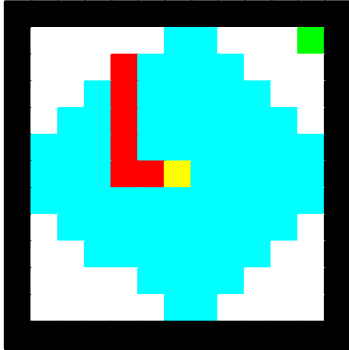
## Grid: BFS 1

---



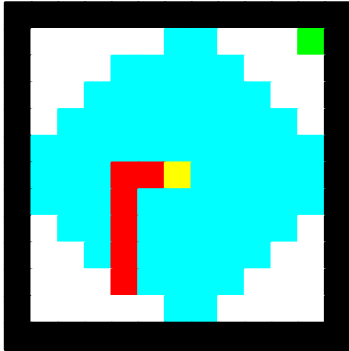
## Grid: BFS 1

---



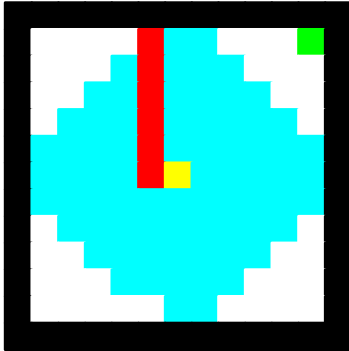
## Grid: BFS 1

---



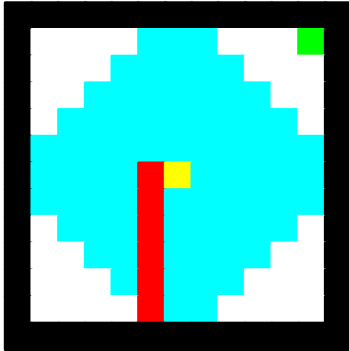
## Grid: BFS 1

---



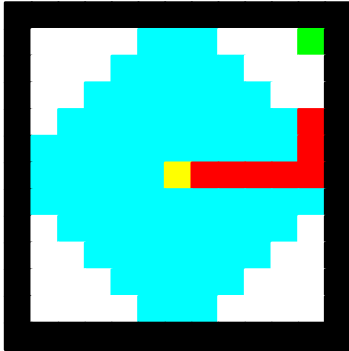
## Grid: BFS 1

---



## Grid: BFS 1

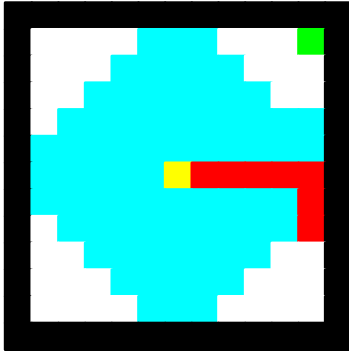
---





## Grid: BFS 1

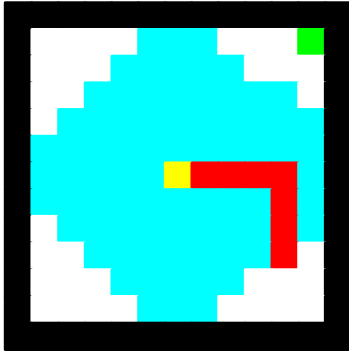
---



**Grid: BFS 1**

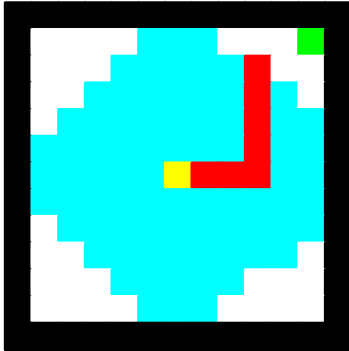
## Grid: BFS 1

---



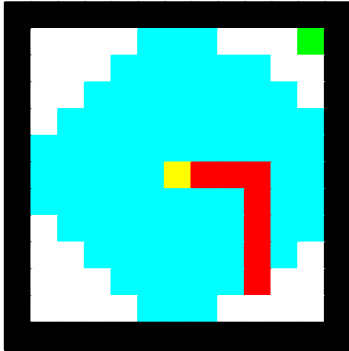
## Grid: BFS 1

---



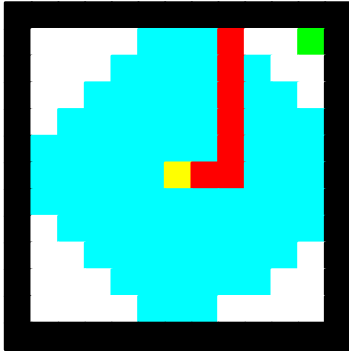
## Grid: BFS 1

---



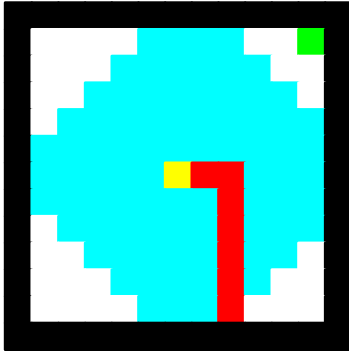
## Grid: BFS 1

---



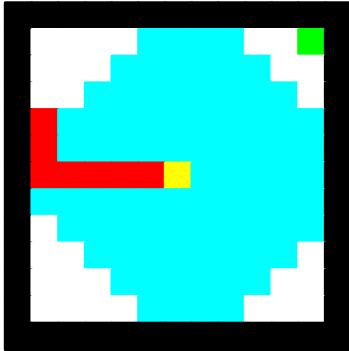
## Grid: BFS 1

---



## Grid: BFS 1

---

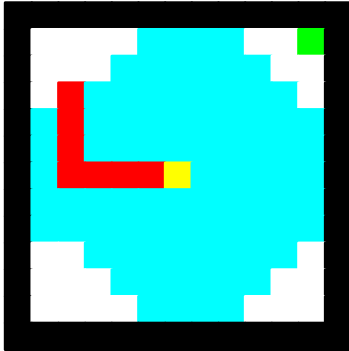




**Grid: BFS 1**

## Grid: BFS 1

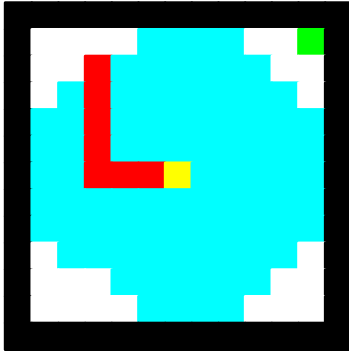
---



**Grid: BFS 1**

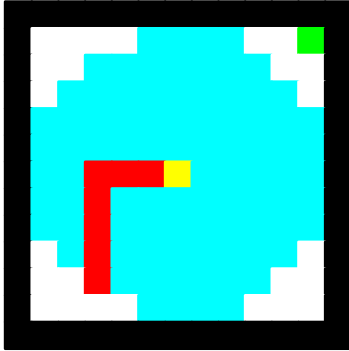
## Grid: BFS 1

---



## Grid: BFS 1

---

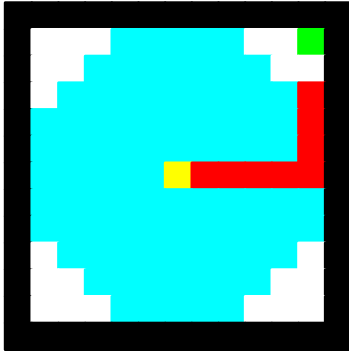


**Grid: BFS 1**

**Grid: BFS 1**

## Grid: BFS 1

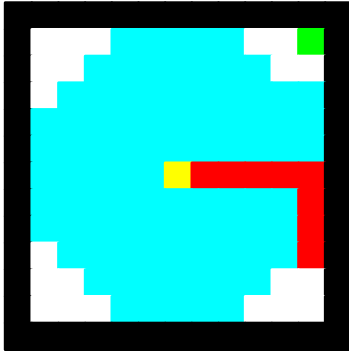
---





## Grid: BFS 1

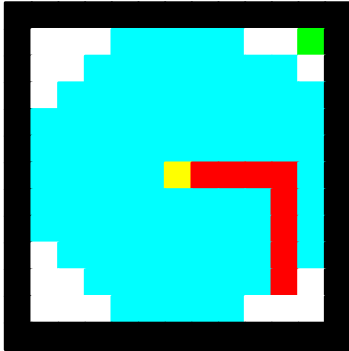
---



**Grid: BFS 1**

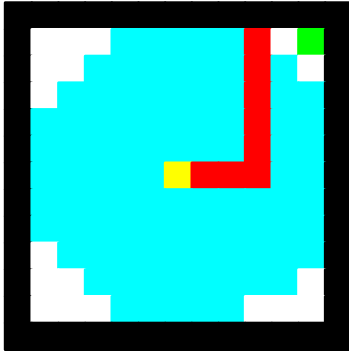
## Grid: BFS 1

---



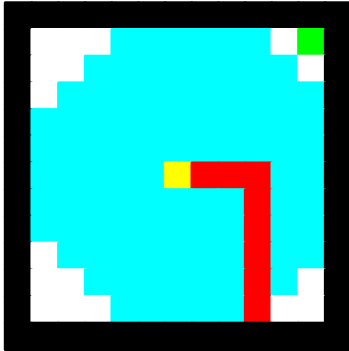
## Grid: BFS 1

---



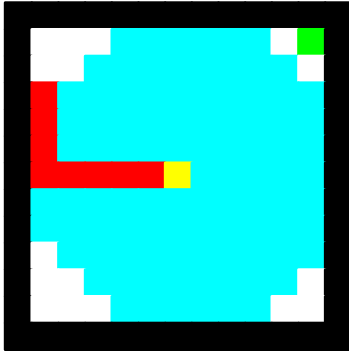
## Grid: BFS 1

---



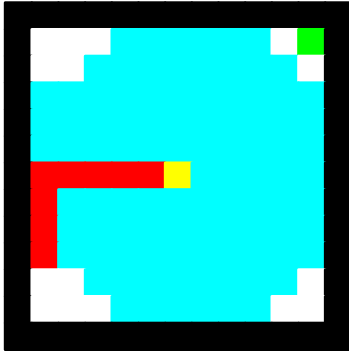
## Grid: BFS 1

---



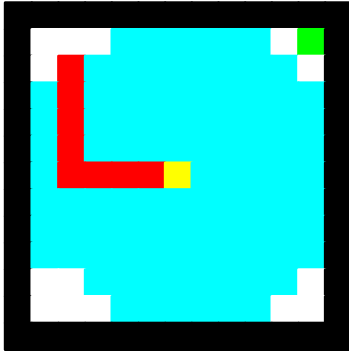
## Grid: BFS 1

---



## Grid: BFS 1

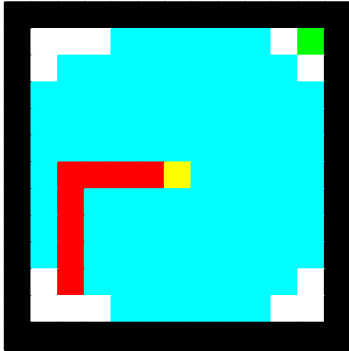
---





## Grid: BFS 1

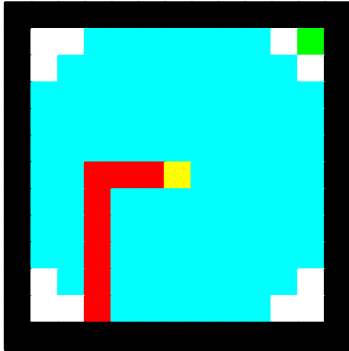
---



**Grid: BFS 1**

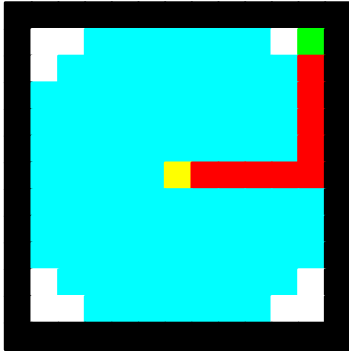
## Grid: BFS 1

---



## Grid: BFS 1

---



**Grid: BFS 1**

**Grid: BFS 1**

**Grid: BFS 1**

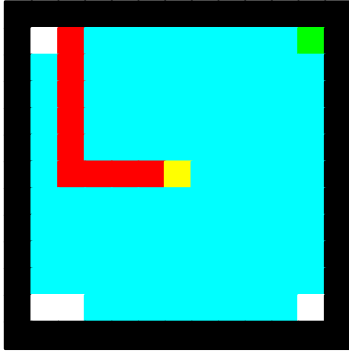
**Grid: BFS 1**



**Grid: BFS 1**

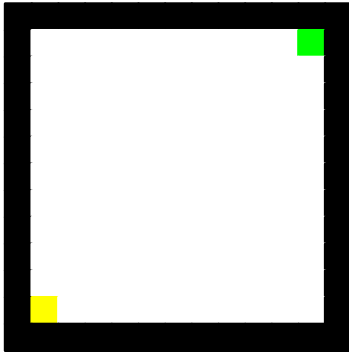
## Grid: BFS 1

---



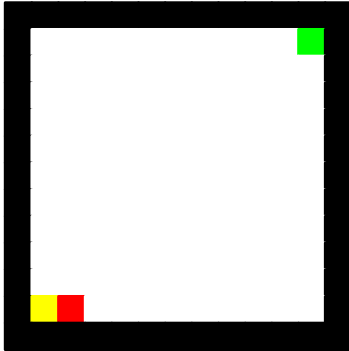
**Grid: BFS 1**

**Grid: BFS 1**



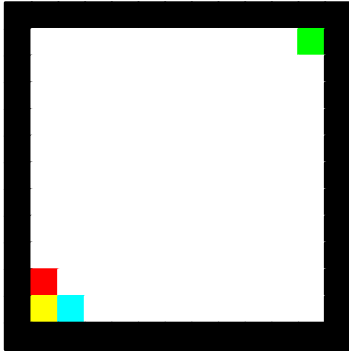
## Grid: BFS 2

---



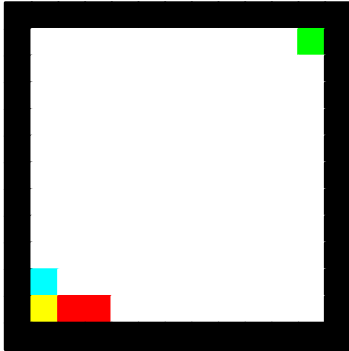
## Grid: BFS 2

---



## Grid: BFS 2

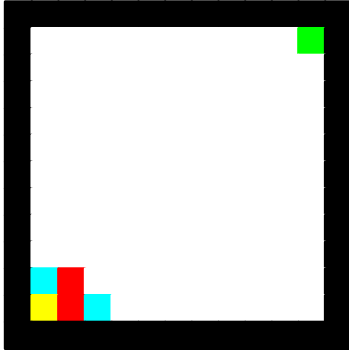
---





## Grid: BFS 2

---

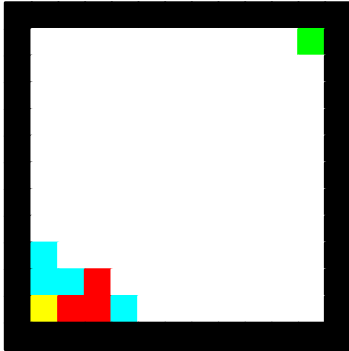


## Grid: BFS 2

## Grid: BFS 2

## Grid: BFS 2

---

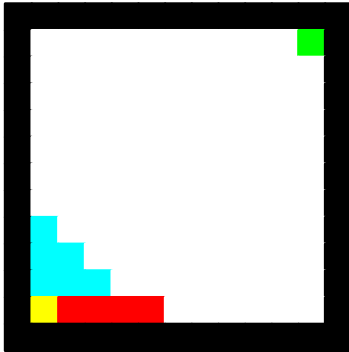


## Grid: BFS 2

## Grid: BFS 2

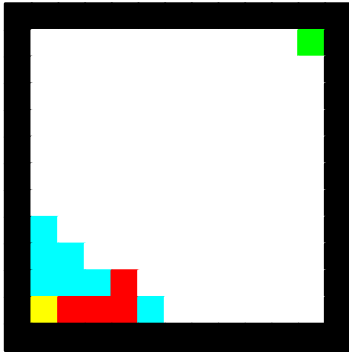
## Grid: BFS 2

---



## Grid: BFS 2

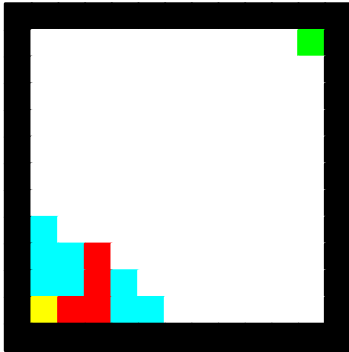
---





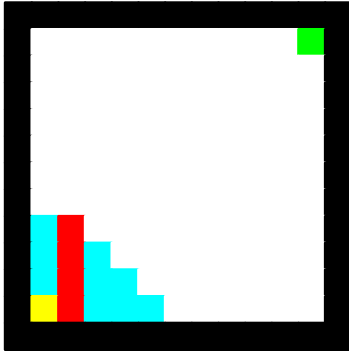
## Grid: BFS 2

---



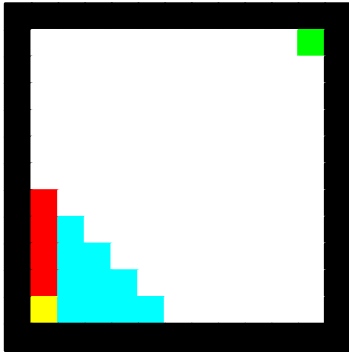
## Grid: BFS 2

---



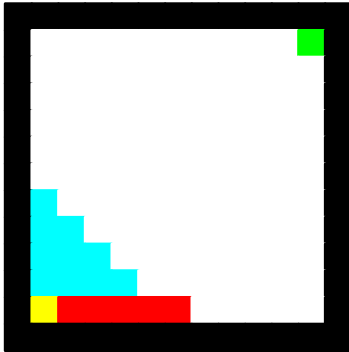
## Grid: BFS 2

---



## Grid: BFS 2

---

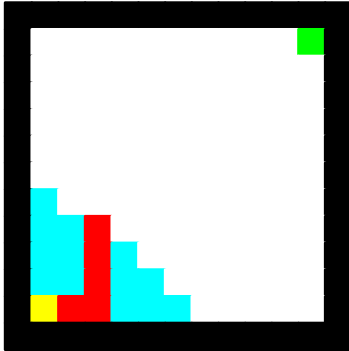


## Grid: BFS 2

## Grid: BFS 2

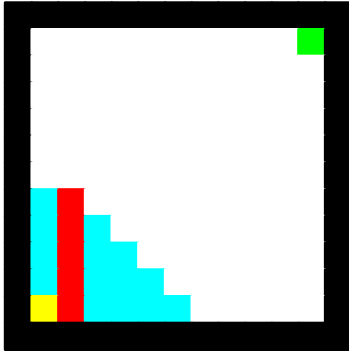
## Grid: BFS 2

---



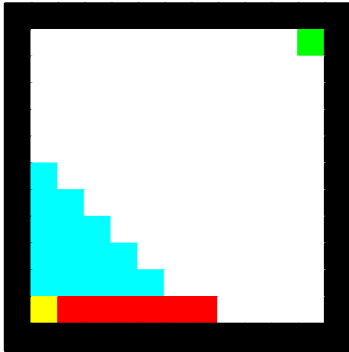
## Grid: BFS 2

---



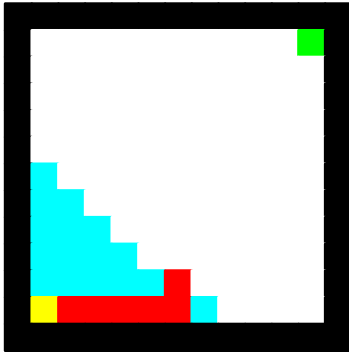


## Grid: BFS 2



## Grid: BFS 2

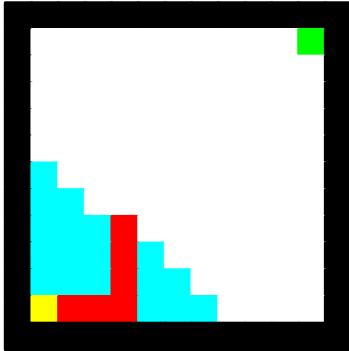
---



## Grid: BFS 2

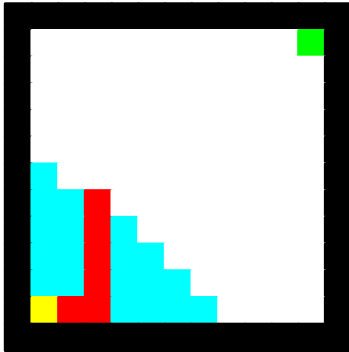
## Grid: BFS 2

---



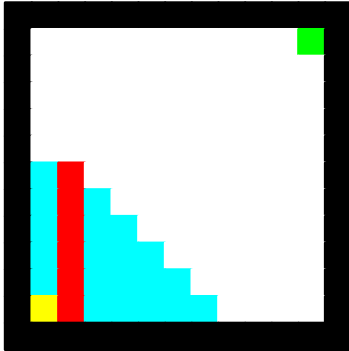
## Grid: BFS 2

---



## Grid: BFS 2

---

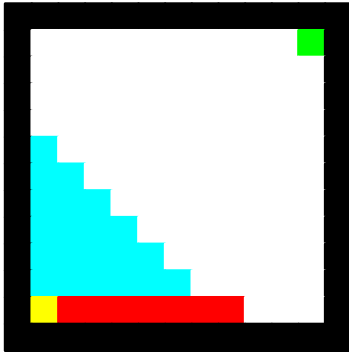


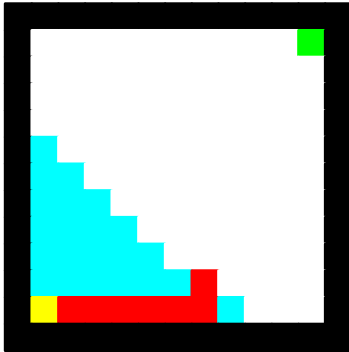
## Grid: BFS 2



## Grid: BFS 2

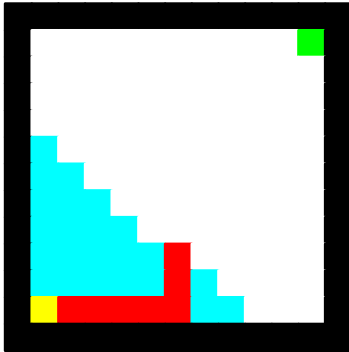
---





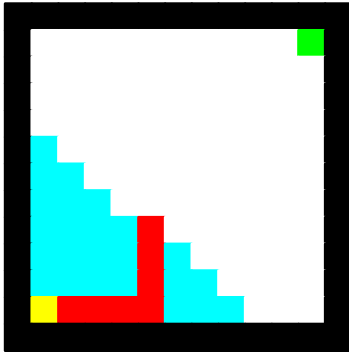
## Grid: BFS 2

---



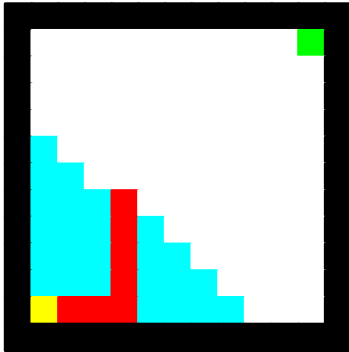
## Grid: BFS 2

---



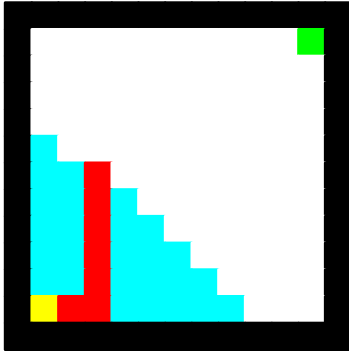
## Grid: BFS 2

---

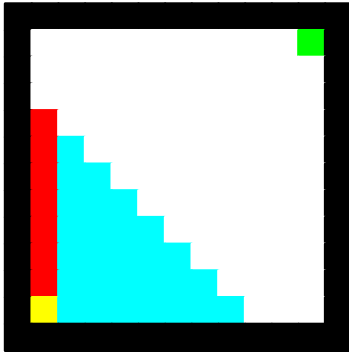


## Grid: BFS 2

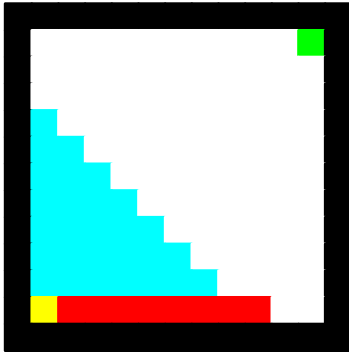
---



## Grid: BFS 2

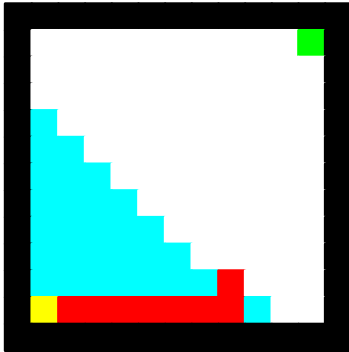






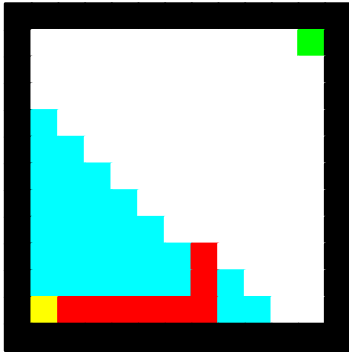
## Grid: BFS 2

---



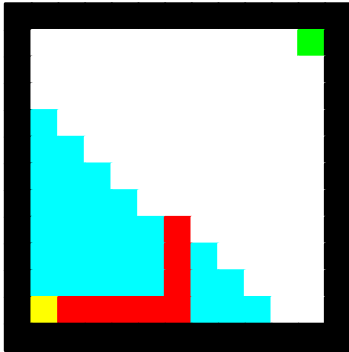
## Grid: BFS 2

---



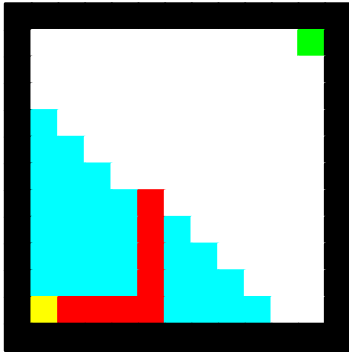
## Grid: BFS 2

---



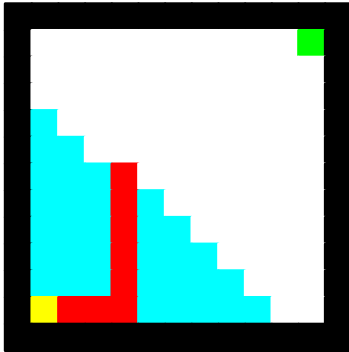
## Grid: BFS 2

---



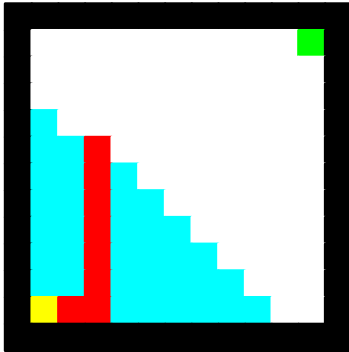
## Grid: BFS 2

---



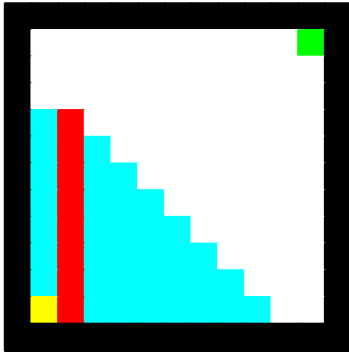
## Grid: BFS 2

---



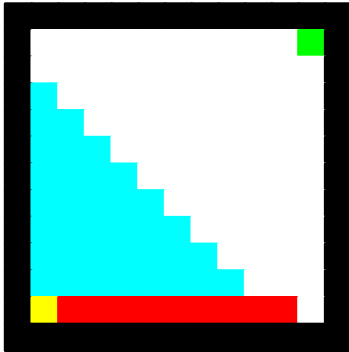
## Grid: BFS 2

---



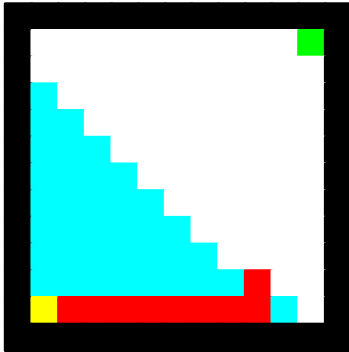


## Grid: BFS 2

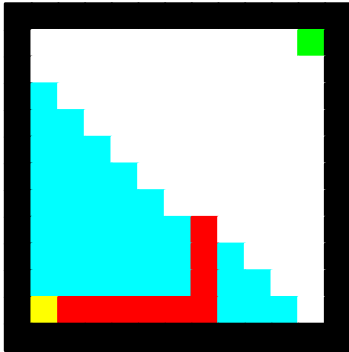


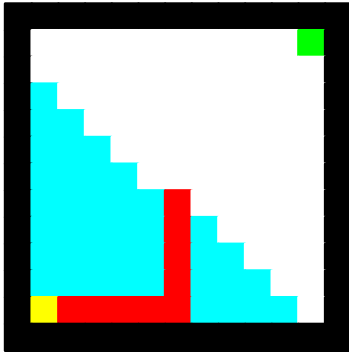
## Grid: BFS 2

---



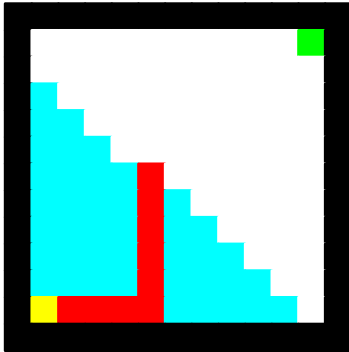
## Grid: BFS 2

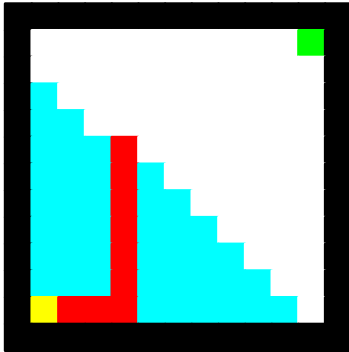




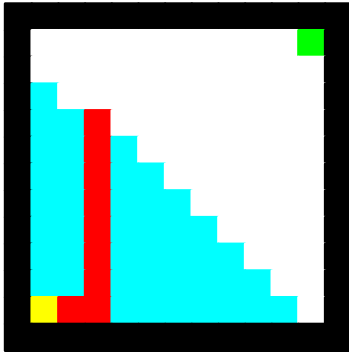
## Grid: BFS 2

---

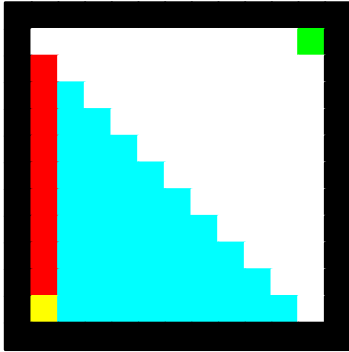


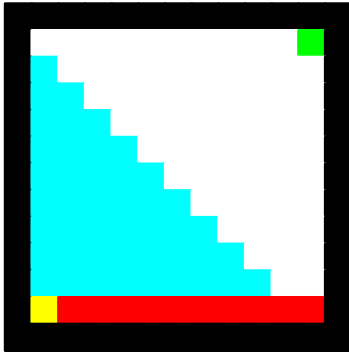






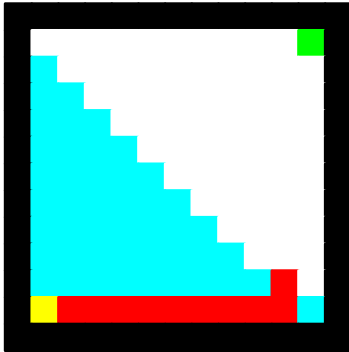






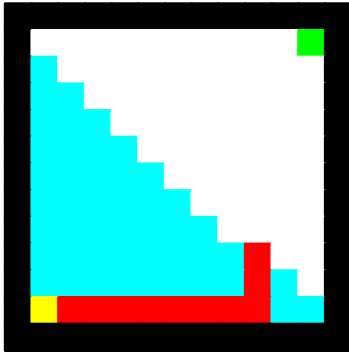
## Grid: BFS 2

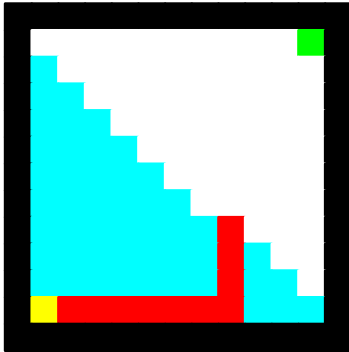
---

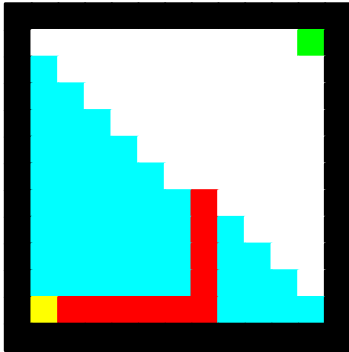


## Grid: BFS 2

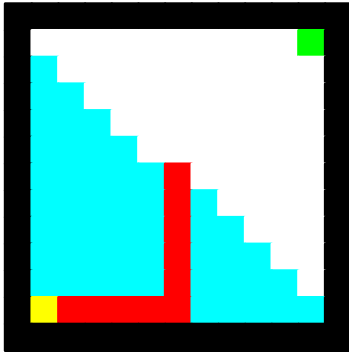
---

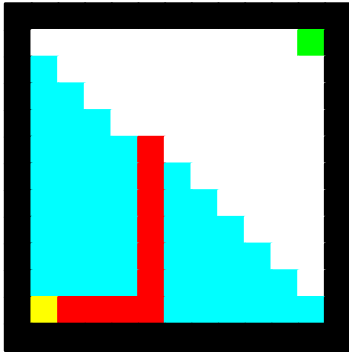




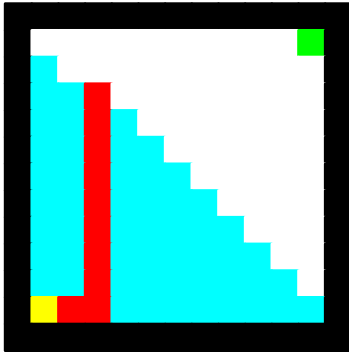


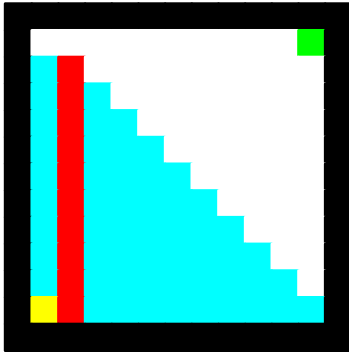






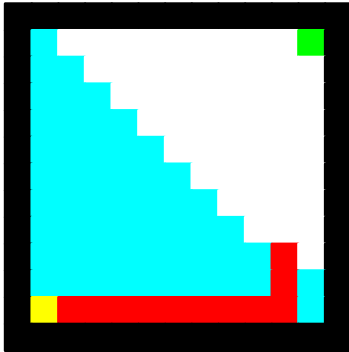






## Grid: BFS 2

## Grid: BFS 2





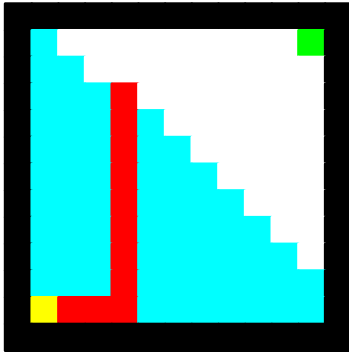
## Grid: BFS 2

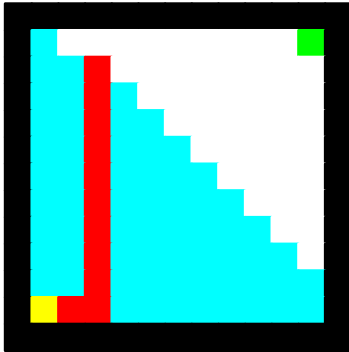
## Grid: BFS 2

## Grid: BFS 2

## Grid: BFS 2

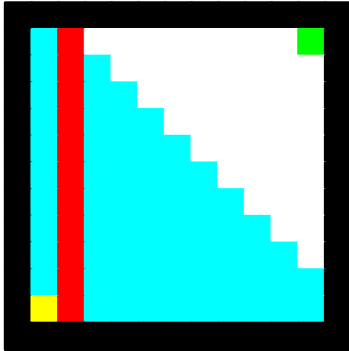
## Grid: BFS 2





## Grid: BFS 2

---

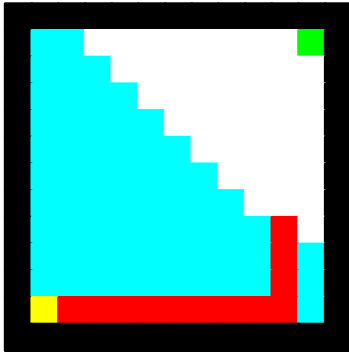




## Grid: BFS 2

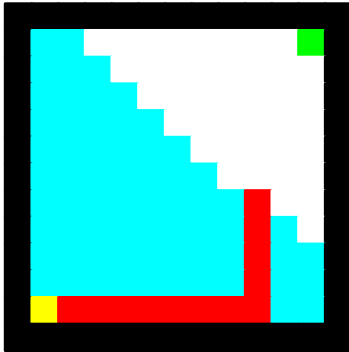
## Grid: BFS 2

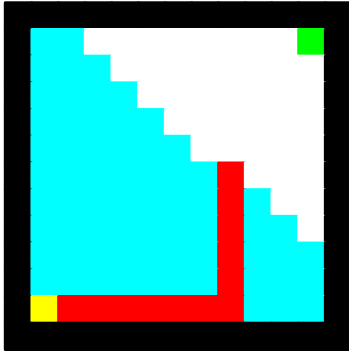
---



## Grid: BFS 2

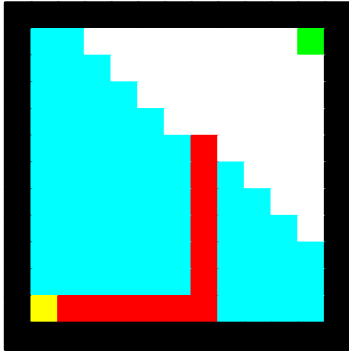
---





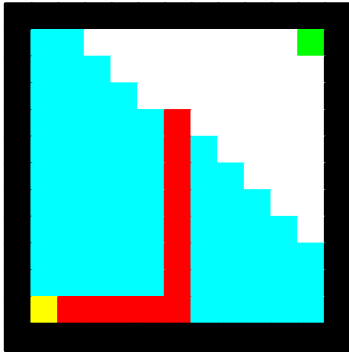
## Grid: BFS 2

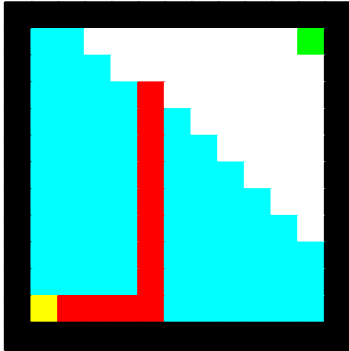
---



## Grid: BFS 2

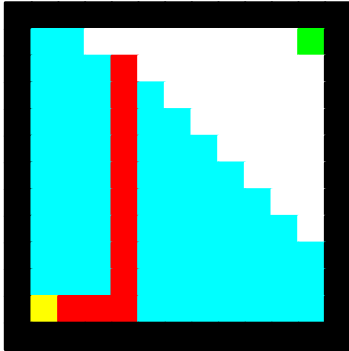
---





## Grid: BFS 2

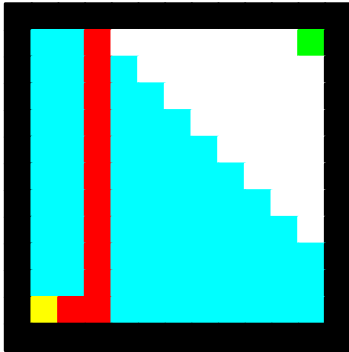
---





## Grid: BFS 2

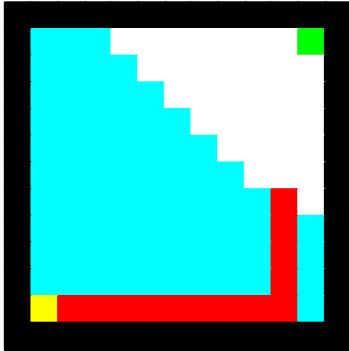
---



## Grid: BFS 2

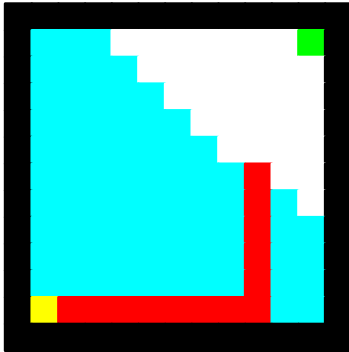
## Grid: BFS 2

---



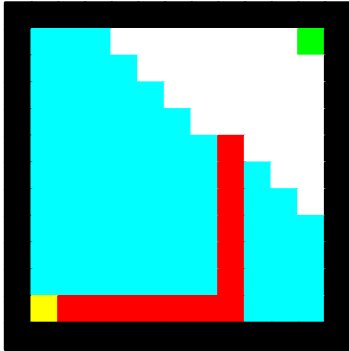
## Grid: BFS 2

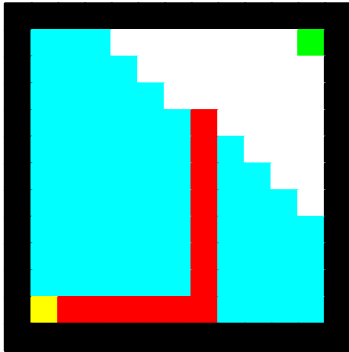
---



## Grid: BFS 2

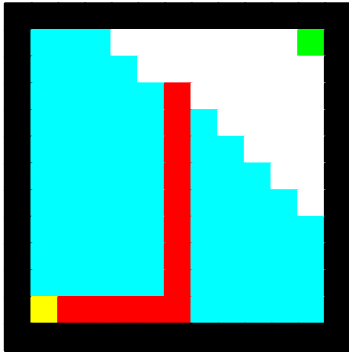
---

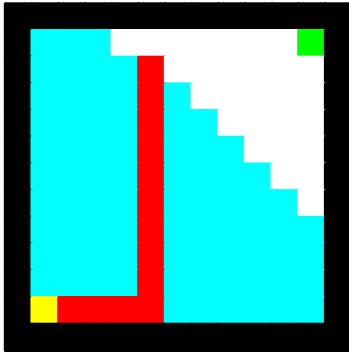




## Grid: BFS 2

---

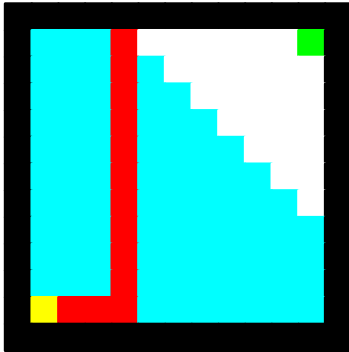






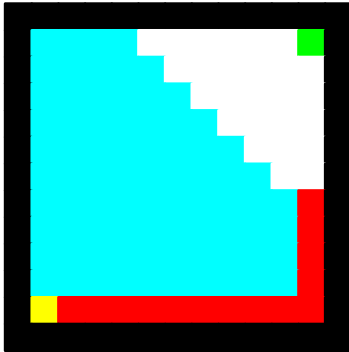
## Grid: BFS 2

---



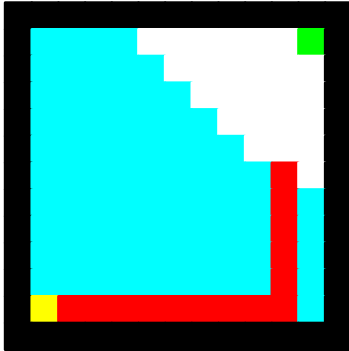
## Grid: BFS 2

---



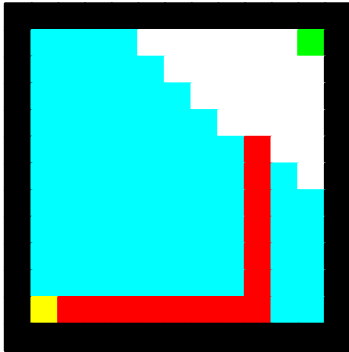
## Grid: BFS 2

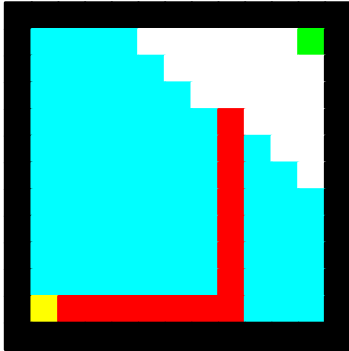
---



## Grid: BFS 2

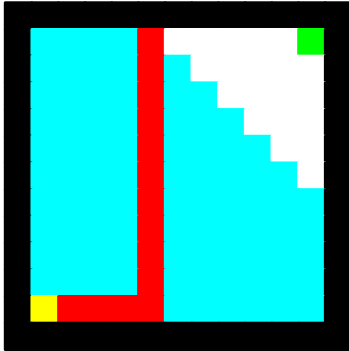
---





## Grid: BFS 2

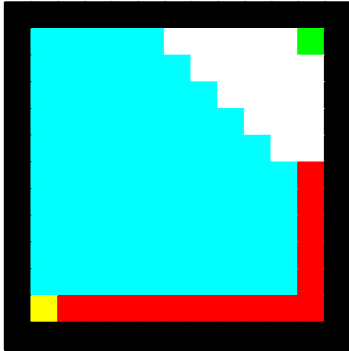
## Grid: BFS 2





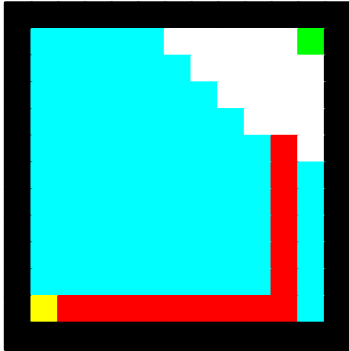
## Grid: BFS 2

---



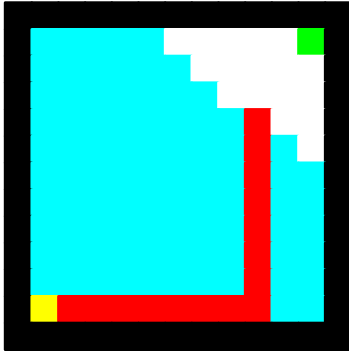
## Grid: BFS 2

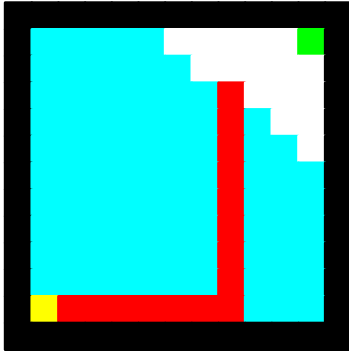
---

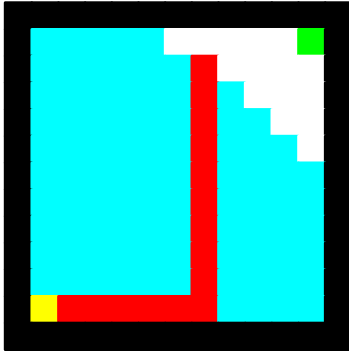


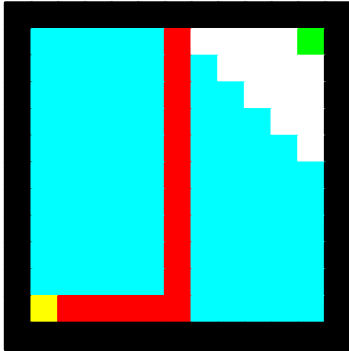
## Grid: BFS 2

---



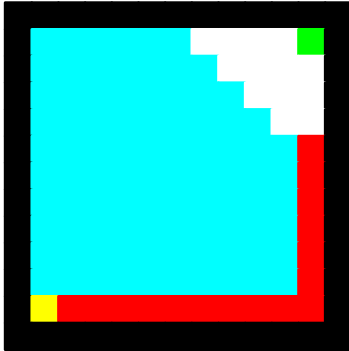


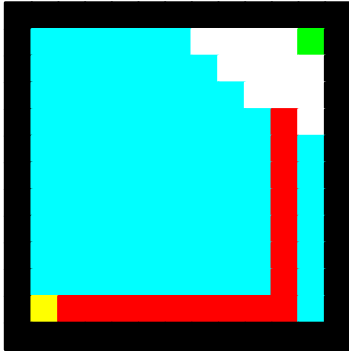




## Grid: BFS 2

---

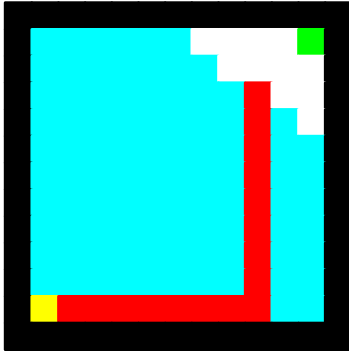






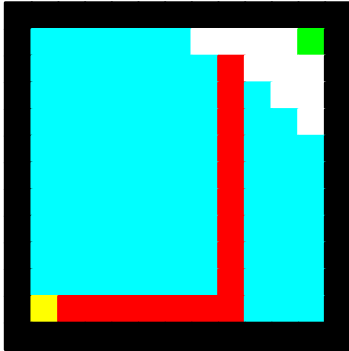
## Grid: BFS 2

---



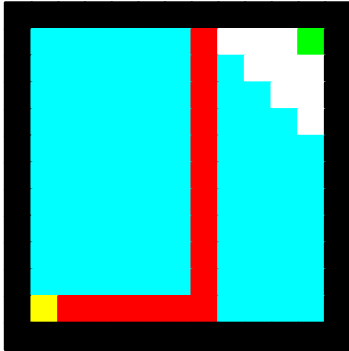
## Grid: BFS 2

---



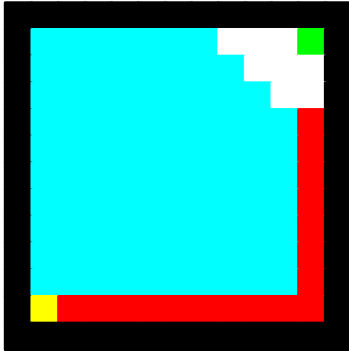
## Grid: BFS 2

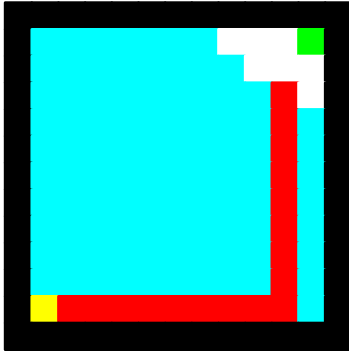
---

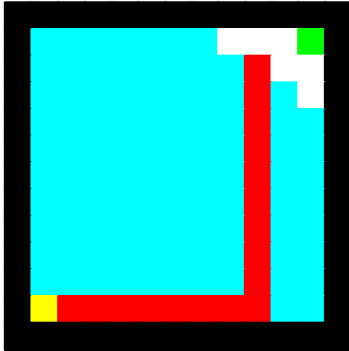


## Grid: BFS 2

---

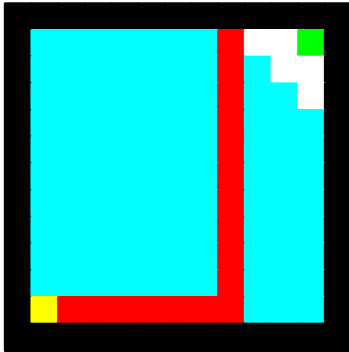






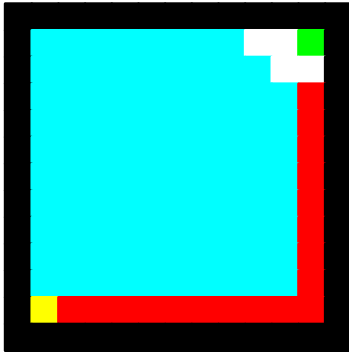
## Grid: BFS 2

---



## Grid: BFS 2

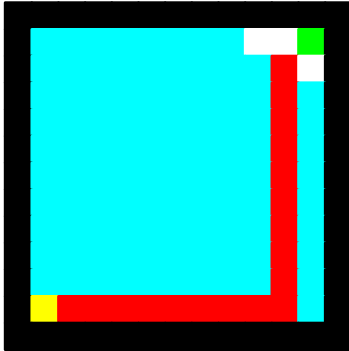
---





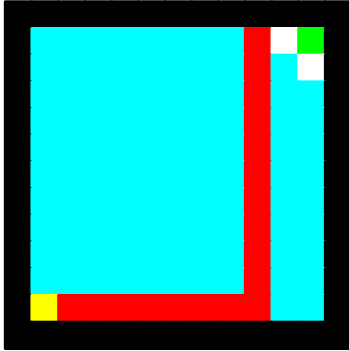
## Grid: BFS 2

---



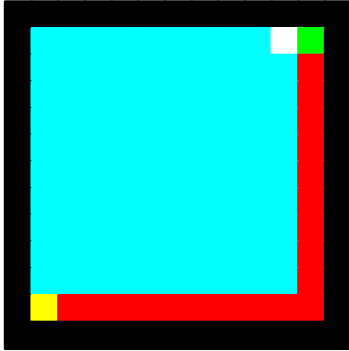
## Grid: BFS 2

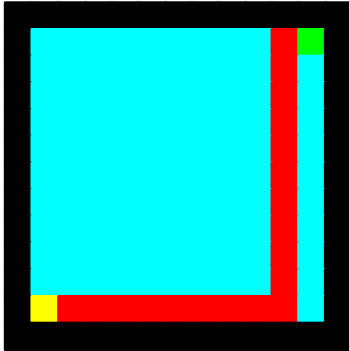
---



## Grid: BFS 2

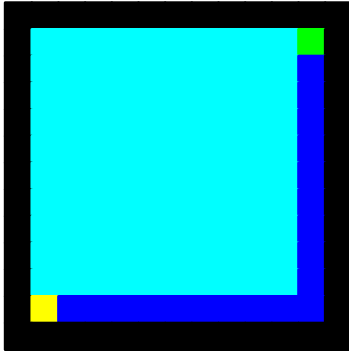
---

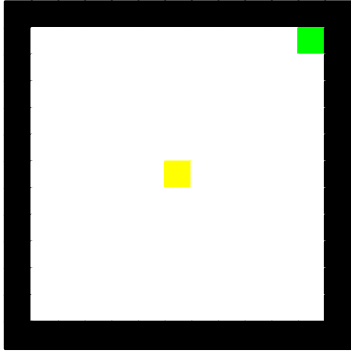


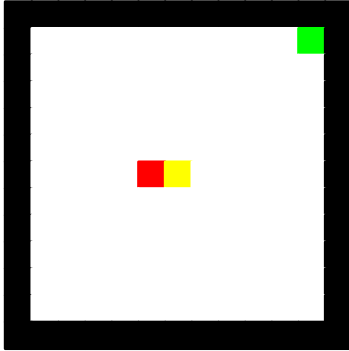


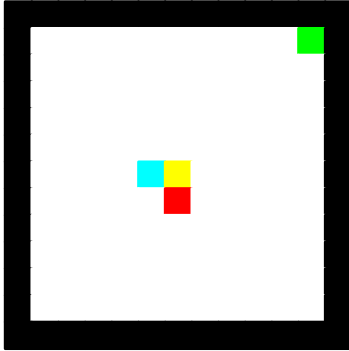
## Grid: BFS 2

---





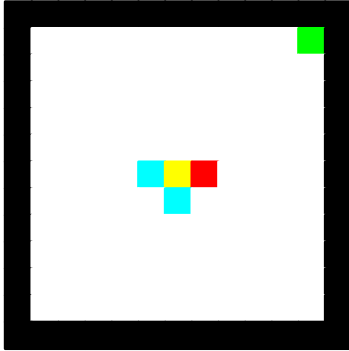


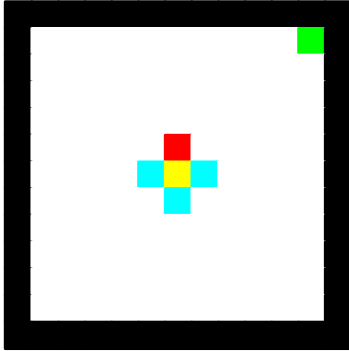


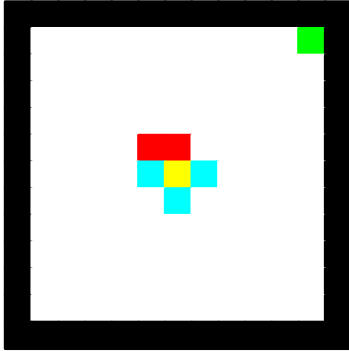


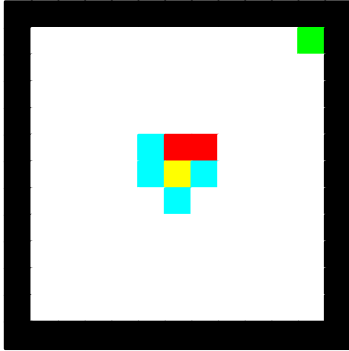
## Grid: DFS 1

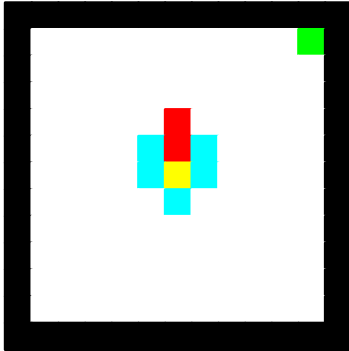
---

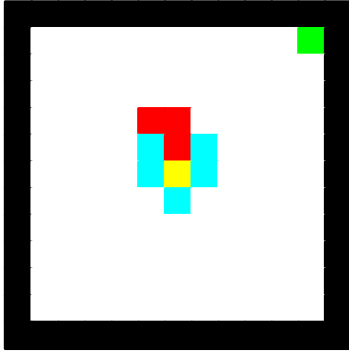




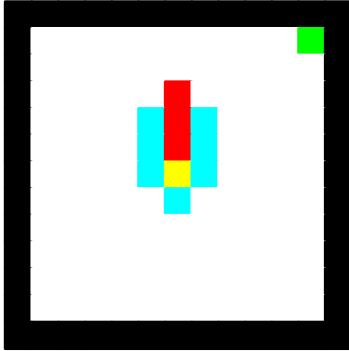




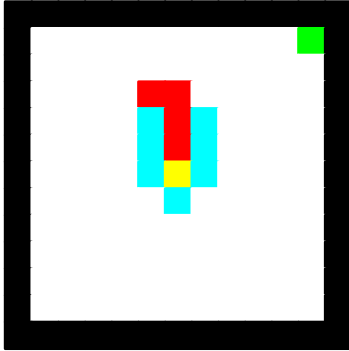






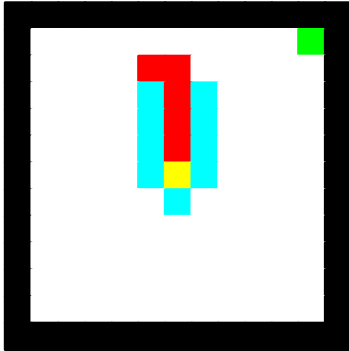


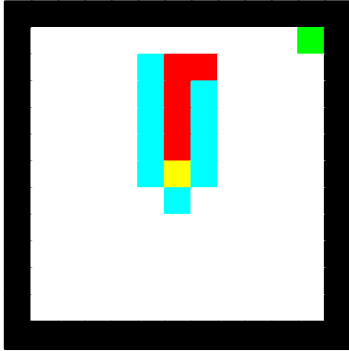


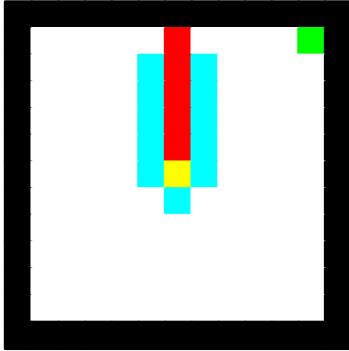




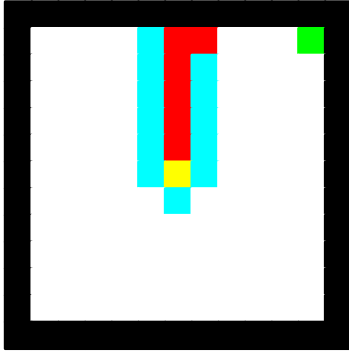


















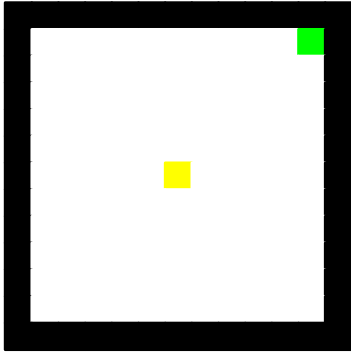




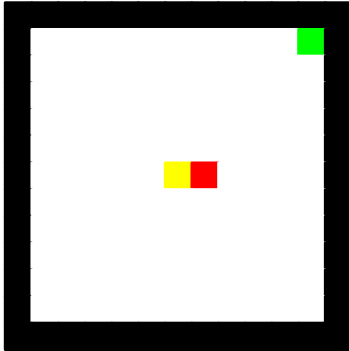


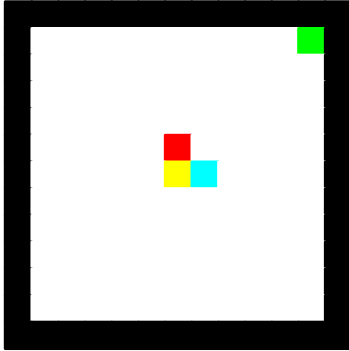




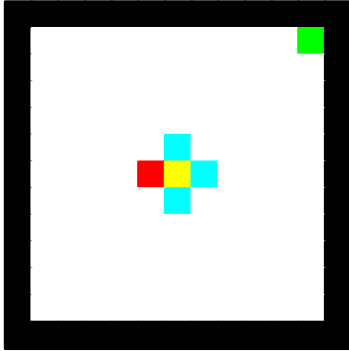


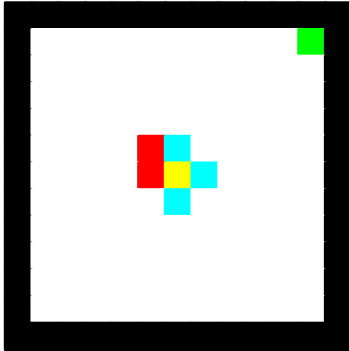


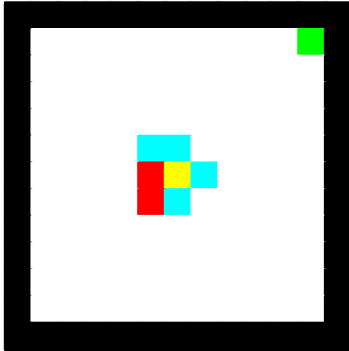


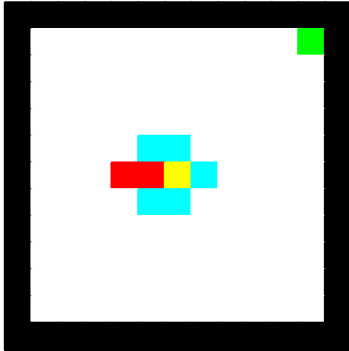








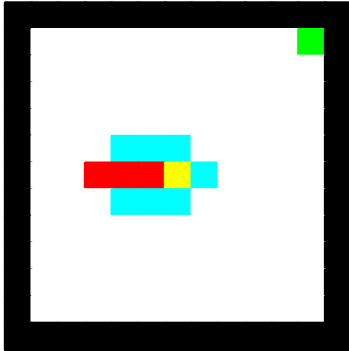


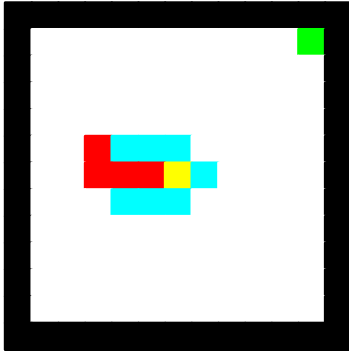


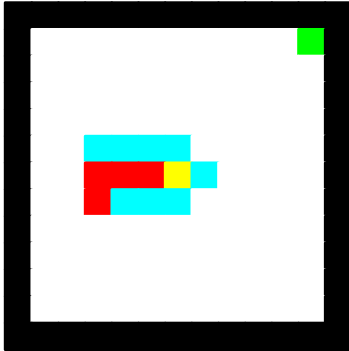


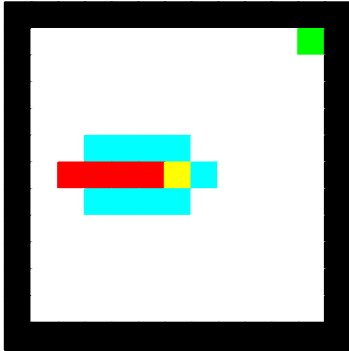


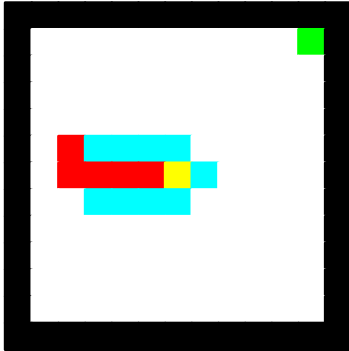


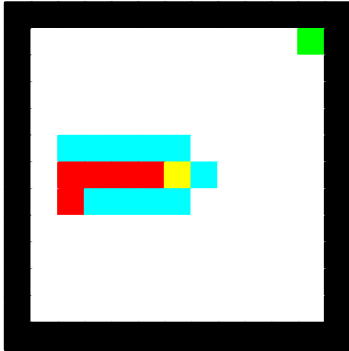


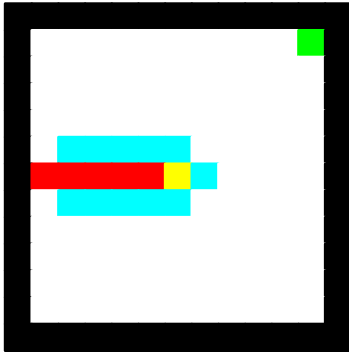






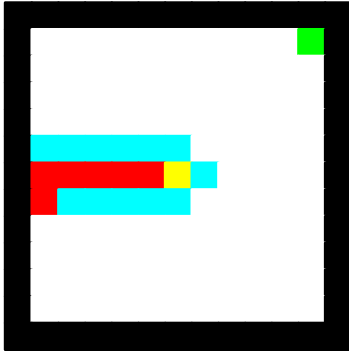






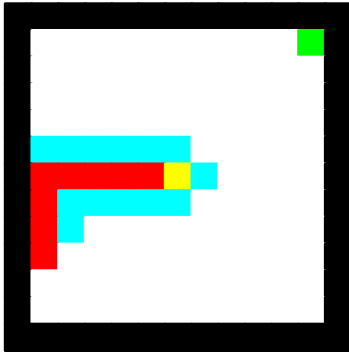










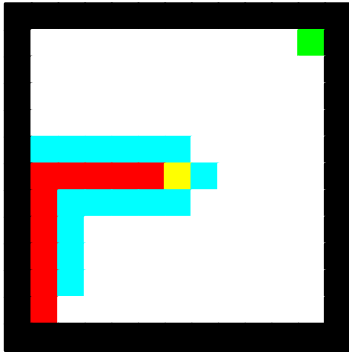


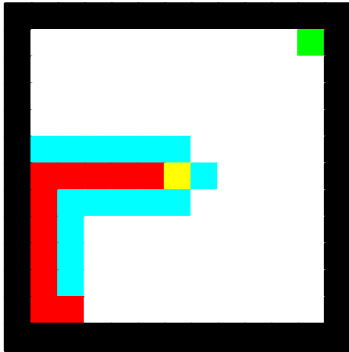






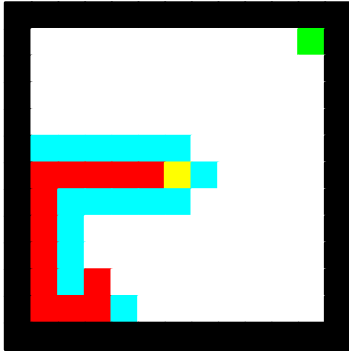




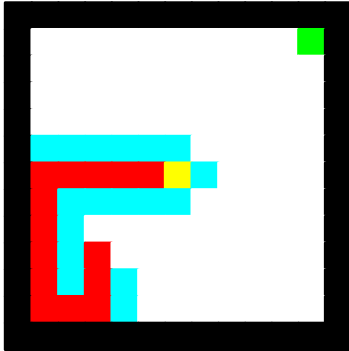


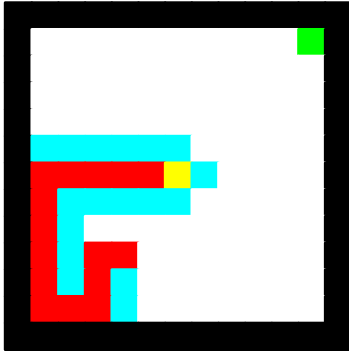






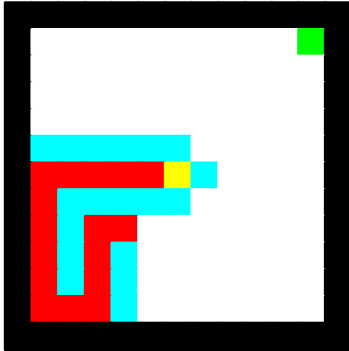


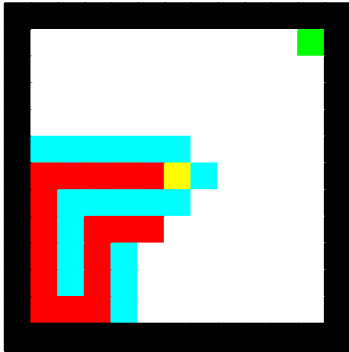




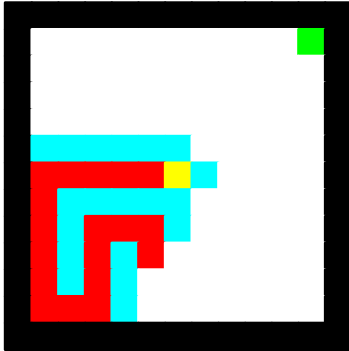






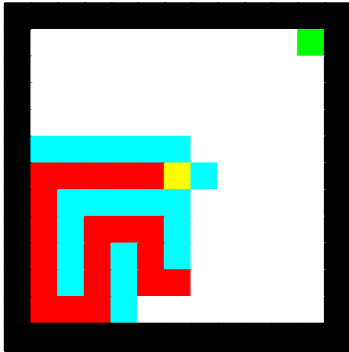




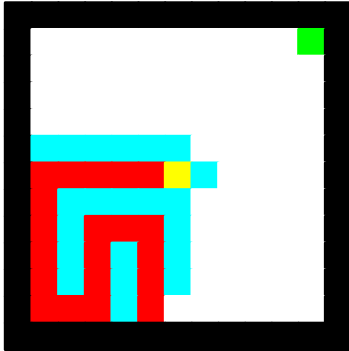




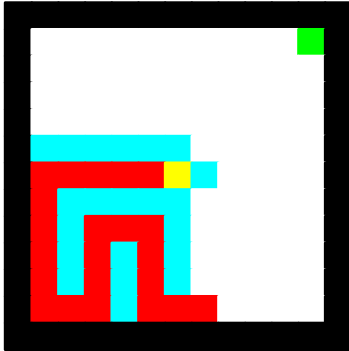


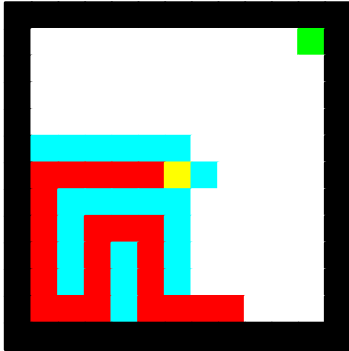


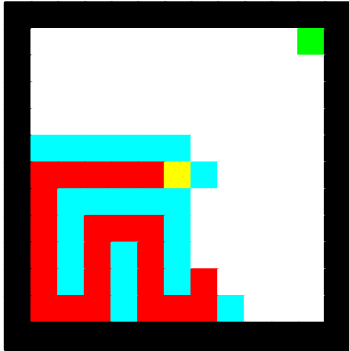




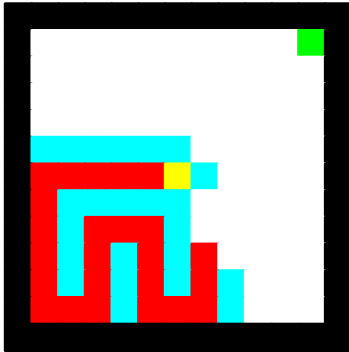














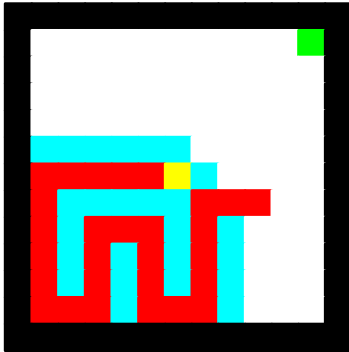


















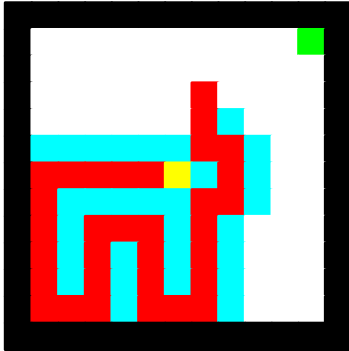


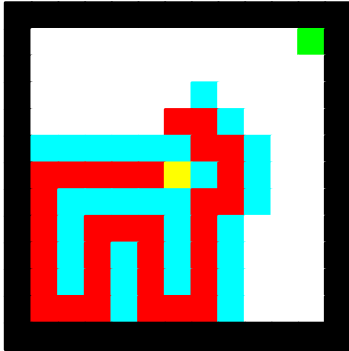


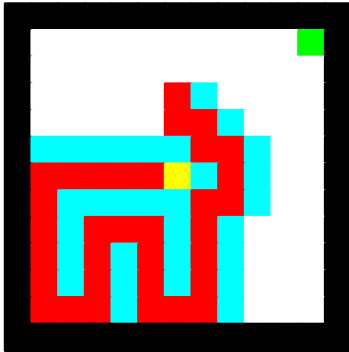


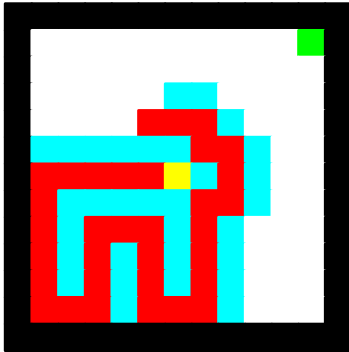




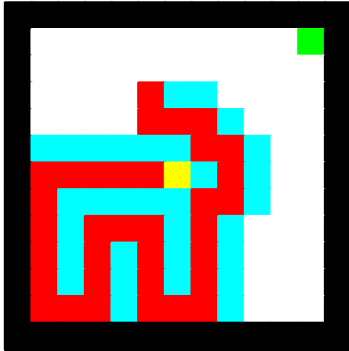










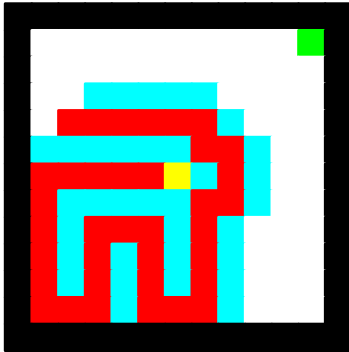








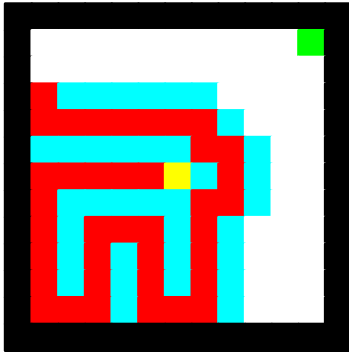


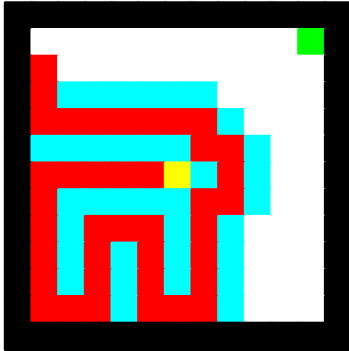


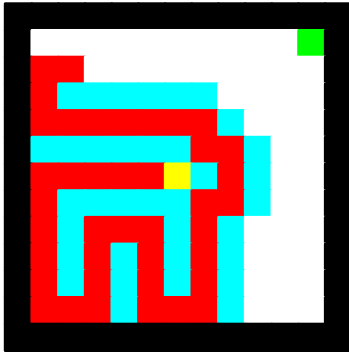






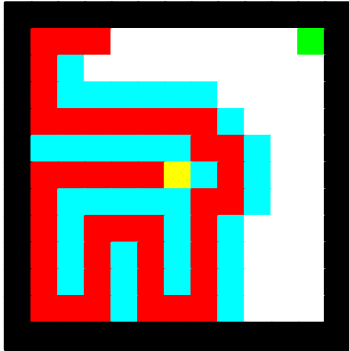


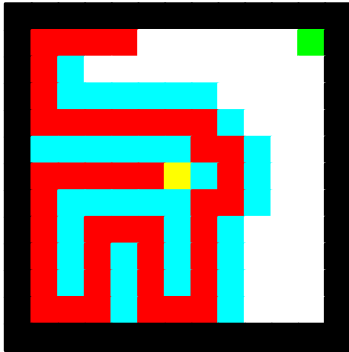


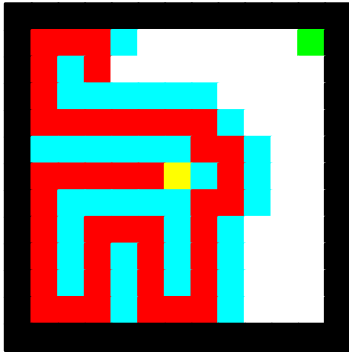








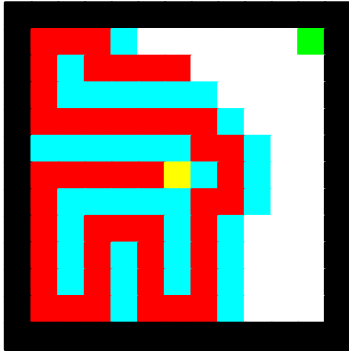


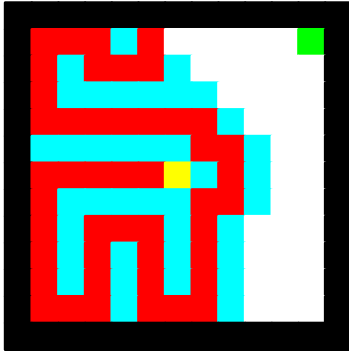




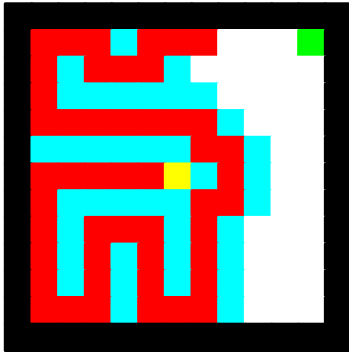


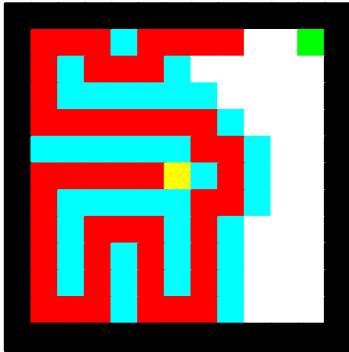


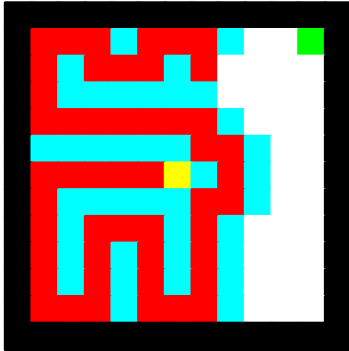




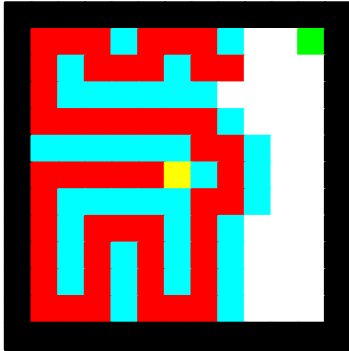


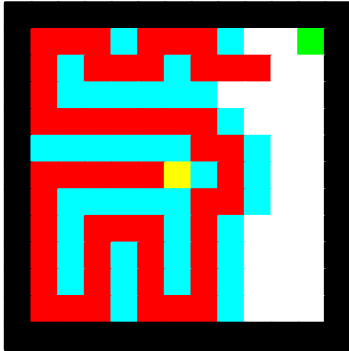


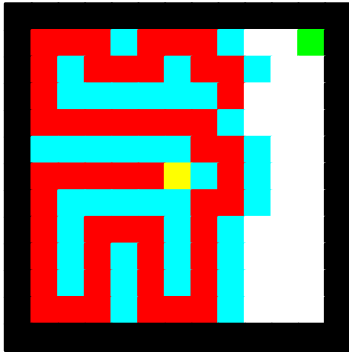


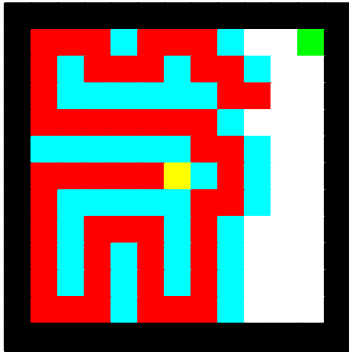


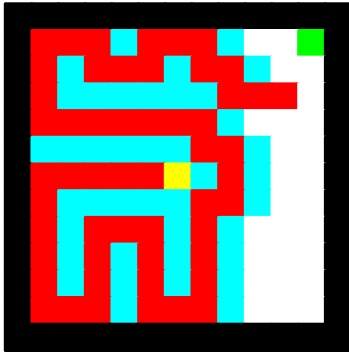


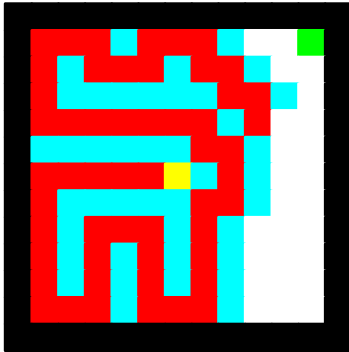


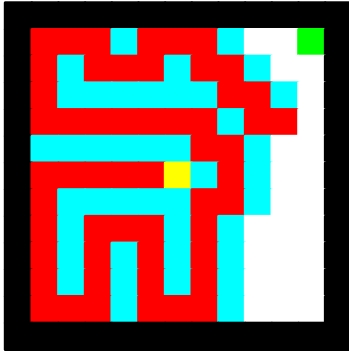






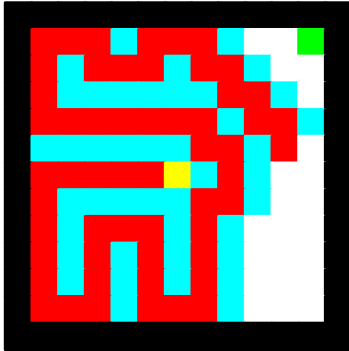


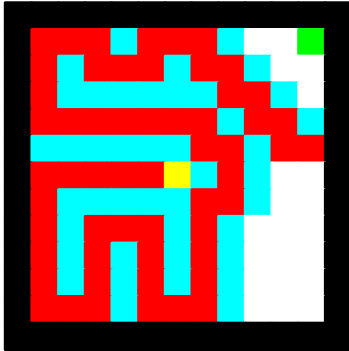








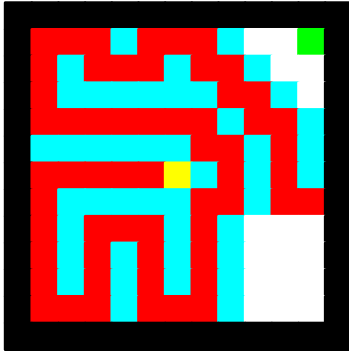








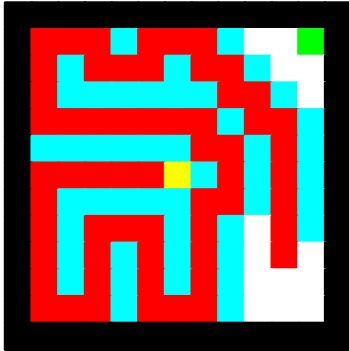


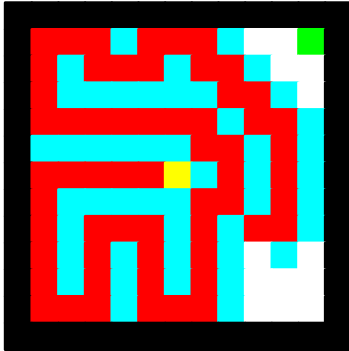


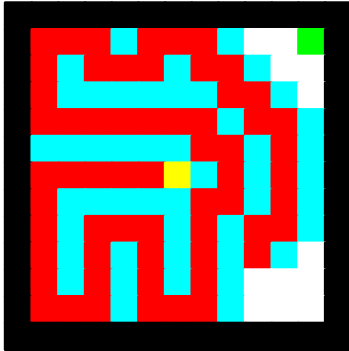


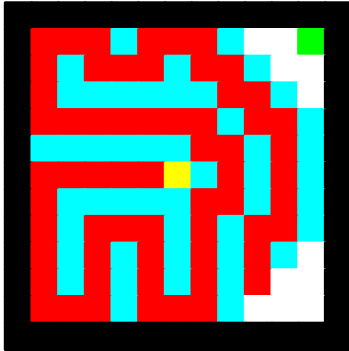


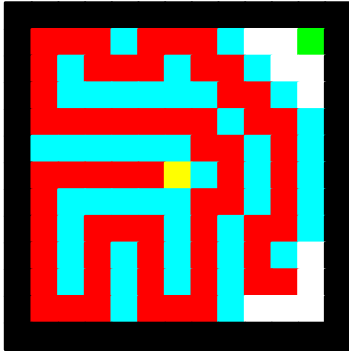


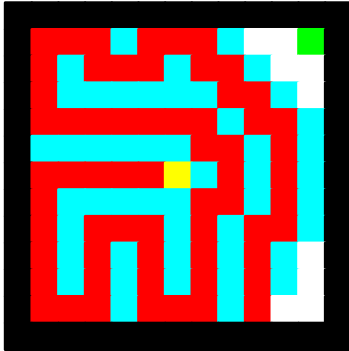


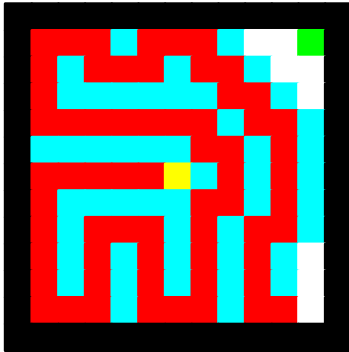








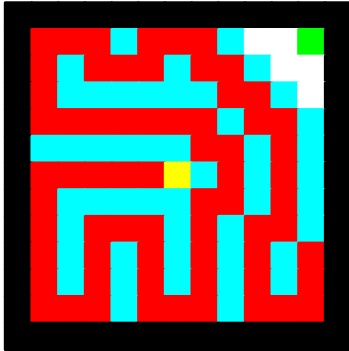


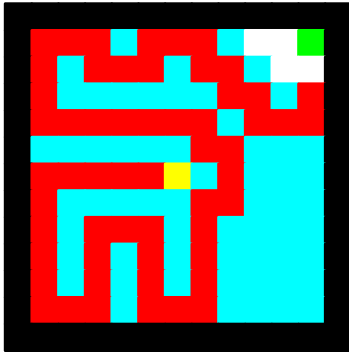




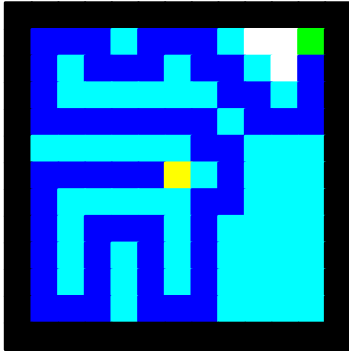












## More Examples

---

## Properties of BFS/DFS

---

### BFS

- Pull and pop from the different sides of the agenda (**f**irst **i**n, **f**irst **o**ut)
- Always returns a shortest path to a goal vertex, if a goal vertex exists in the set of vertices reachable from the start vertex.
- May run forever in an infinite domain if there is no solution.
- Generally requires more memory than depth-first search.

### DFS

- Pull and pop from the same of the agenda (**l**ast **i**n, **f**irst **o**ut)
- Always returns **a** path to a goal vertex, if a goal vertex exists in the set of vertices reachable from the start vertex **and the search domain is finite**.
- May run forever in an infinite domain.
- Doesn't necessarily find the shortest path.
- Efficient in the amount of space it requires to store the agenda.

## Adding Costs

---

BFS is guaranteed always to return the shortest possible path (in terms of total number of edges traversed) to the goal.

Suppose that some edges are more beneficial than others to traverse, though. BFS's sense of optimality does not take that information into account.

Next, we'll develop a new algorithm that takes into account a sense of how desirable each edge is.



## What is a Graph?

---

Set  $V$  of vertices (or "states")

Set  $E$  of edges connecting vertices

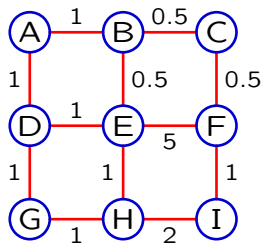
Set  $W$  of edge costs (or "weights")

New goal: rather than finding the *shortest* path, we want to find a path with minimum total *cost*

## Example

---

Find path  $E \rightarrow I$ , minimizing total cost



## Uniform-Cost Search

---

Consider searching for **least-cost** paths instead of *shortest* paths. Instead of popping from agenda based on when nodes were added, pop based on of the cost of the paths they represent.

By considering paths in order of increasing cost, we guarantee that we return a path to the goal that has minimal cost.

## Uniform-Cost Search

---

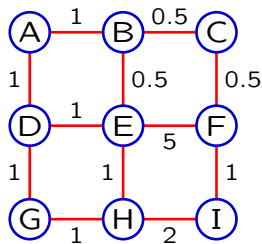
Uniform Cost Search:

- Initialize **expanded set** (all vertices that have been *removed* from the agenda)
  - Initialize agenda (list of paths to consider)
  - Repeat the following:
    - Remove one path from the agenda
    - **If its terminal vertex is in the expanded set, ignore it.**
    - **If its terminal vertex satisfies the goal condition, return a path.**
    - **Add its terminal vertex to the expanded set**
    - Add each child to the agenda if its terminal vertex is **not already in the expanded set**
- until **goal is found** or **agenda is empty**

## Example

---

Find path  $E \rightarrow I$ , minimizing total cost



## More Examples

---

## More Examples

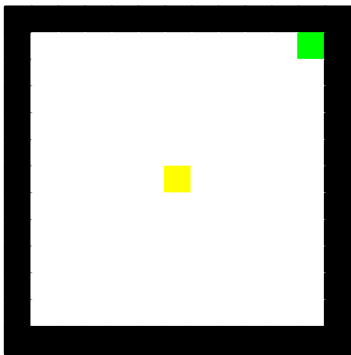
---

## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



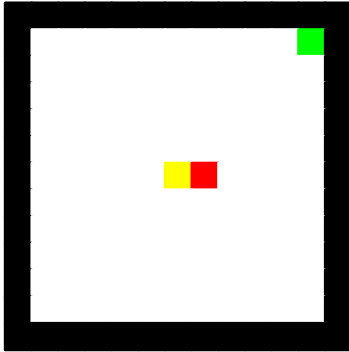


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

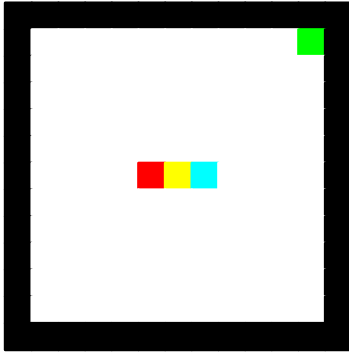


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

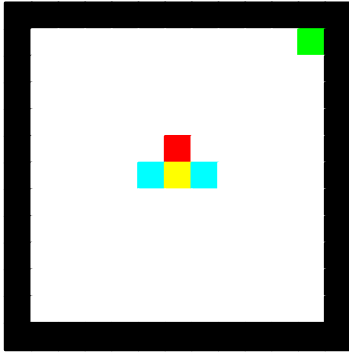


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

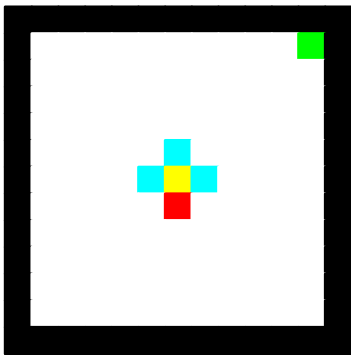


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

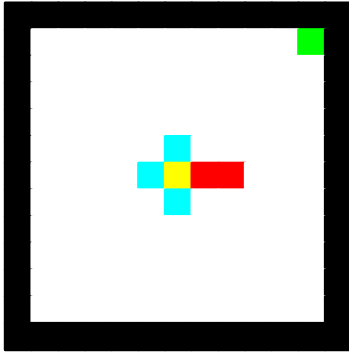


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

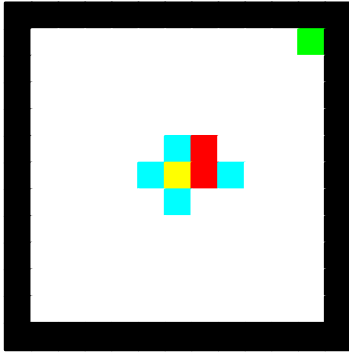


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

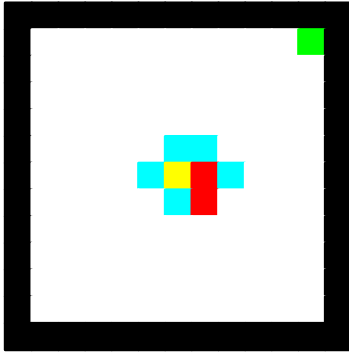


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

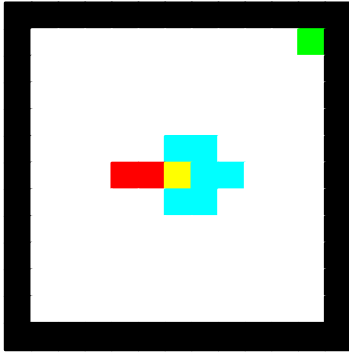


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



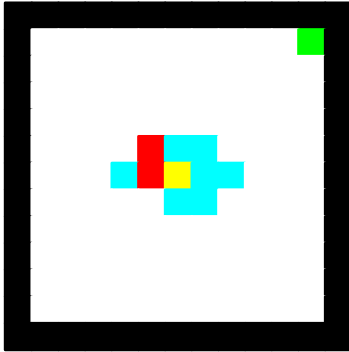


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

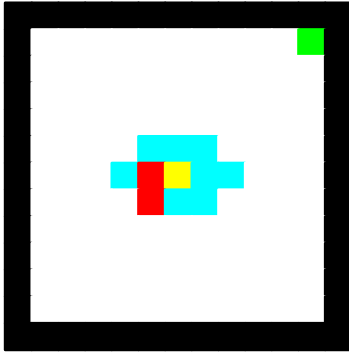


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

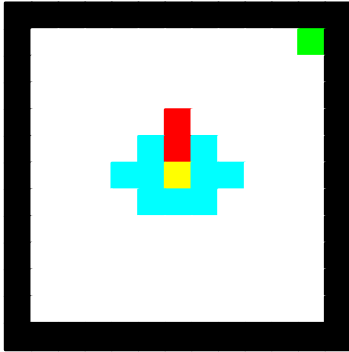


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

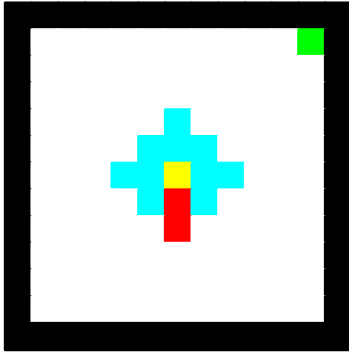


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

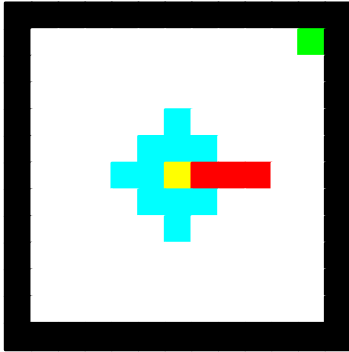


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

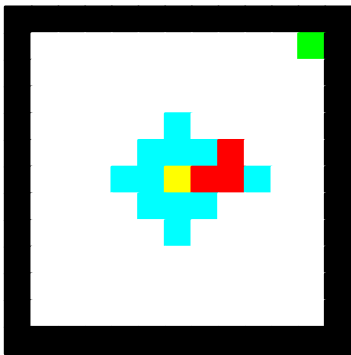


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

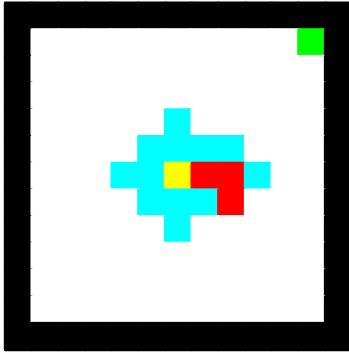


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

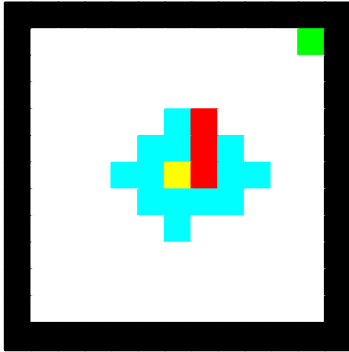


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



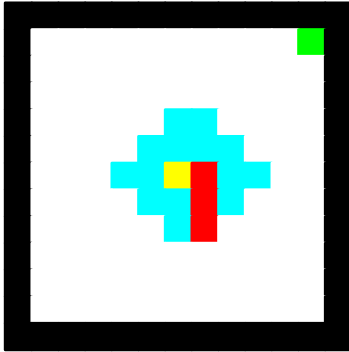


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

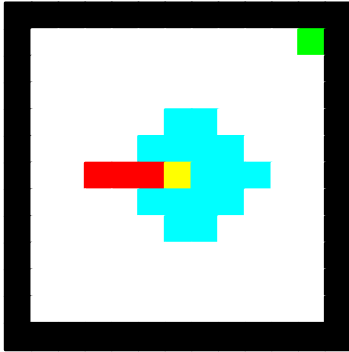


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

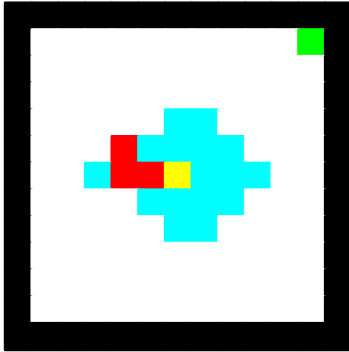


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

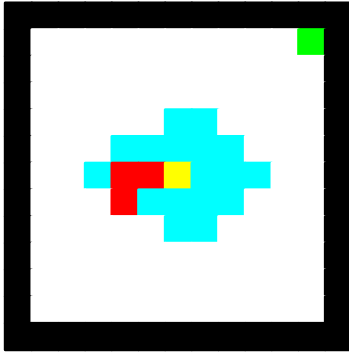


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

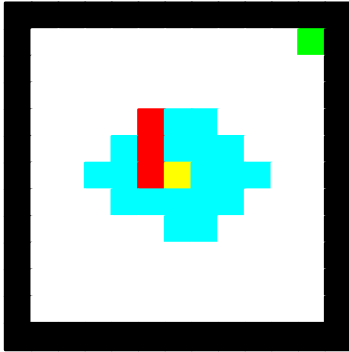


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

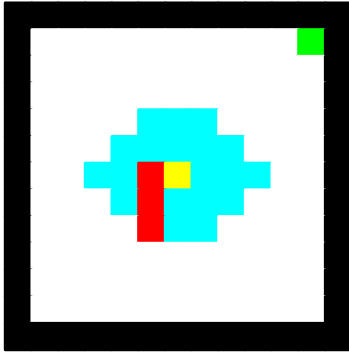


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

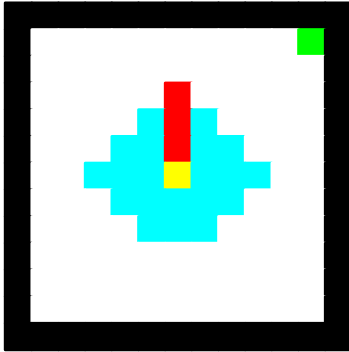


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

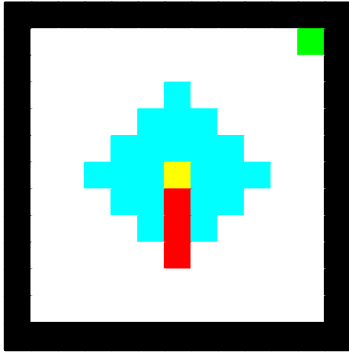


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



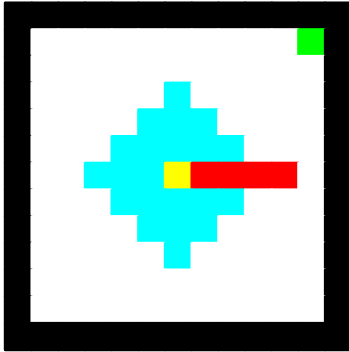


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

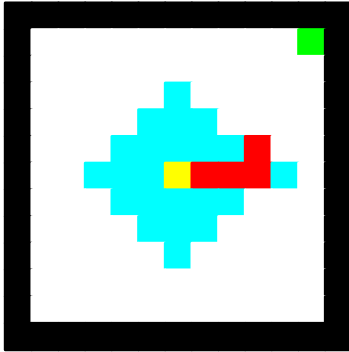


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

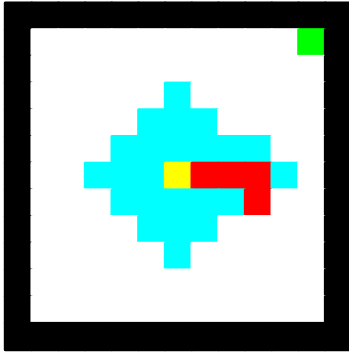


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

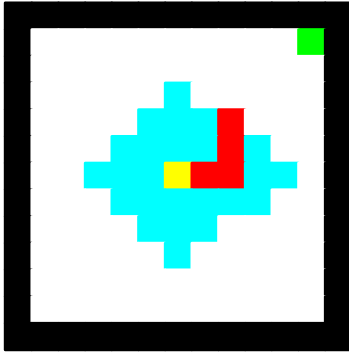


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

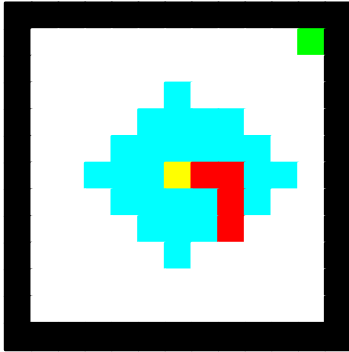


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

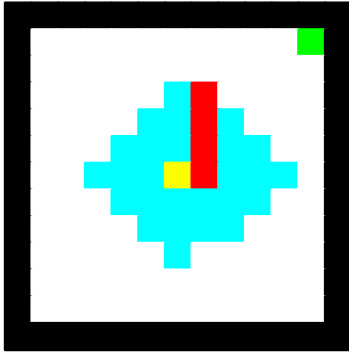


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

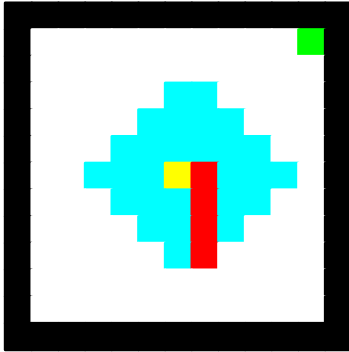


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

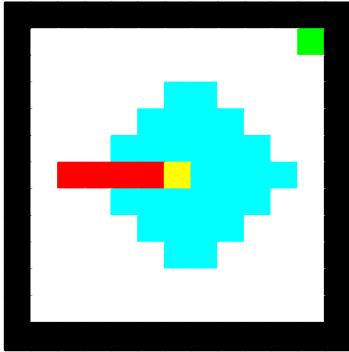


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



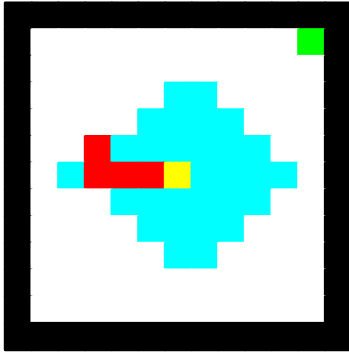


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

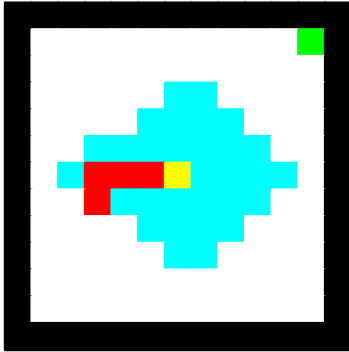


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

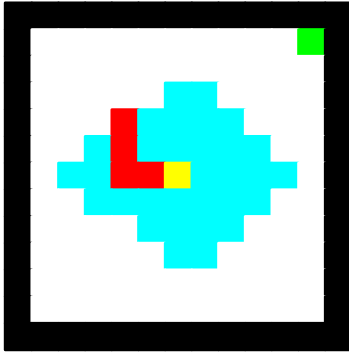


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

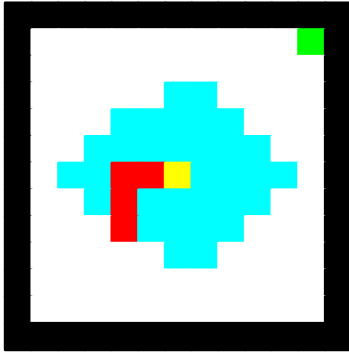


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

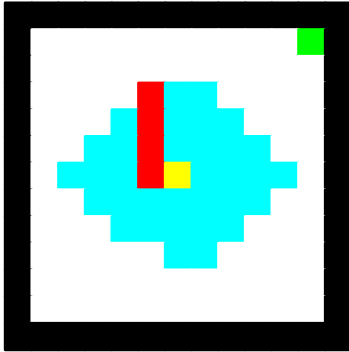


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

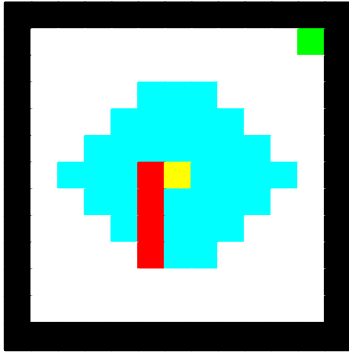


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

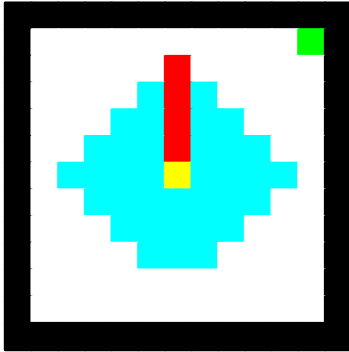


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

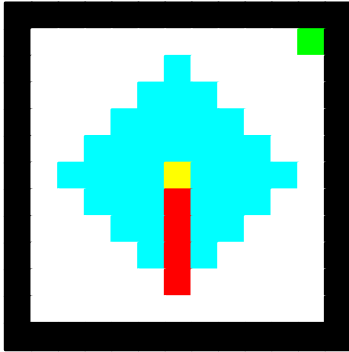


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



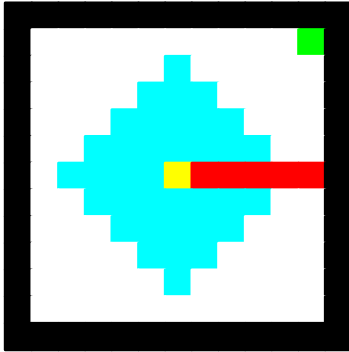


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

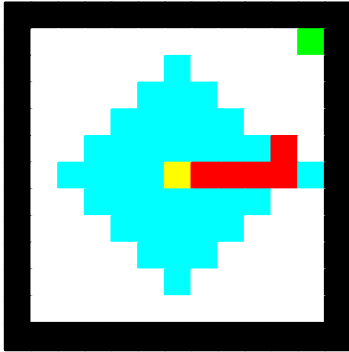


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

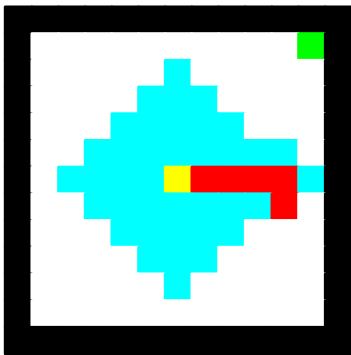


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

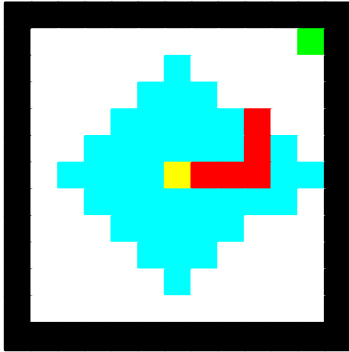


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

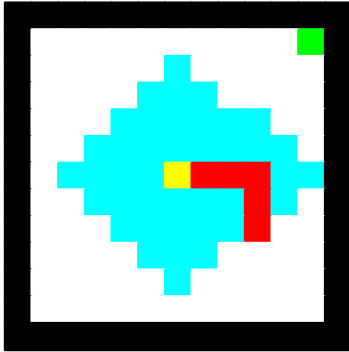


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

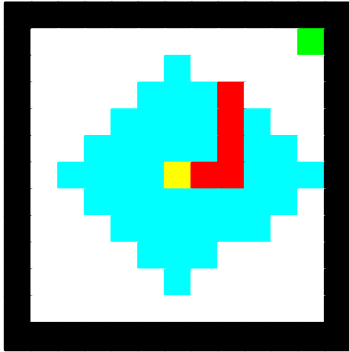


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

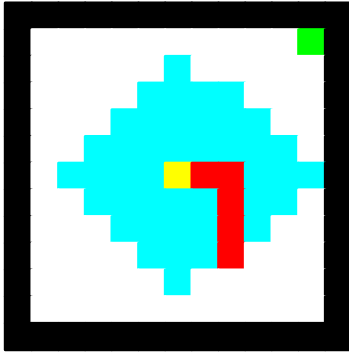


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

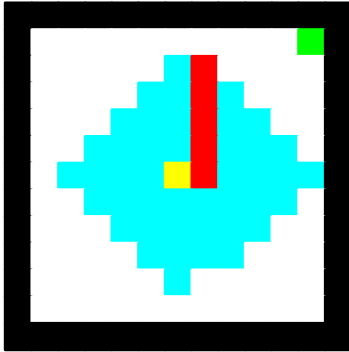


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



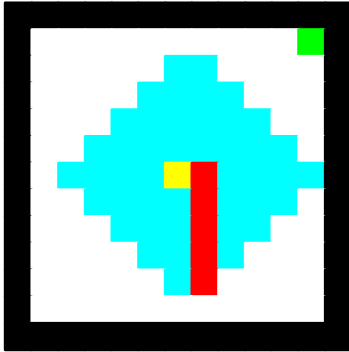


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

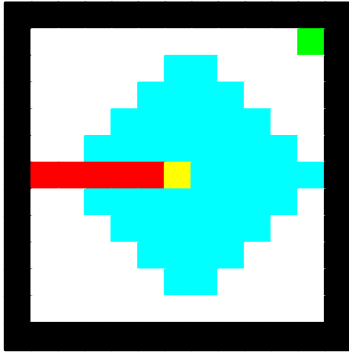


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

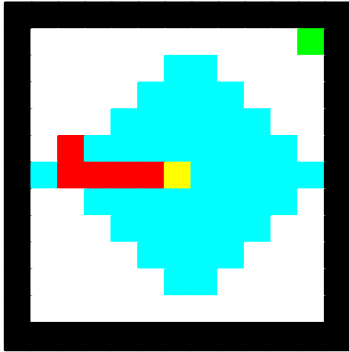


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

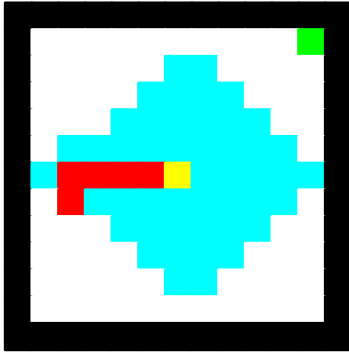


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

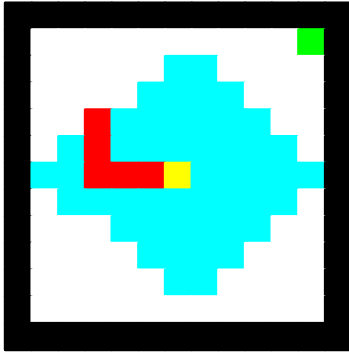


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

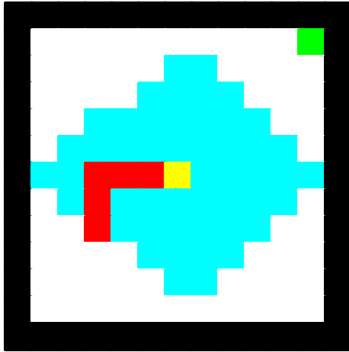


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

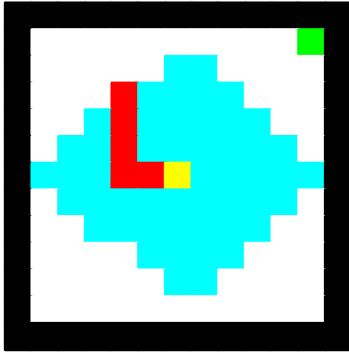


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

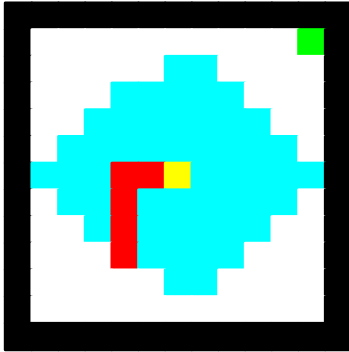


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



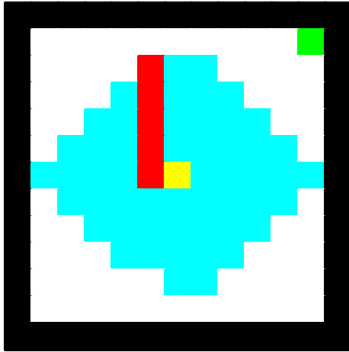


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

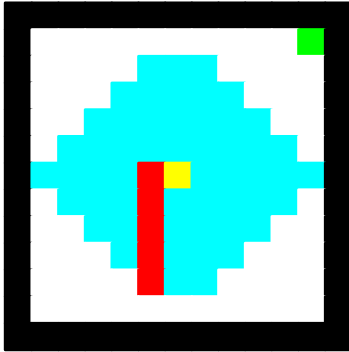


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

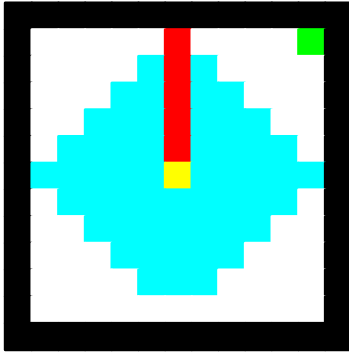


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

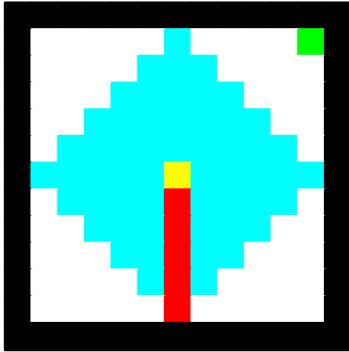


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

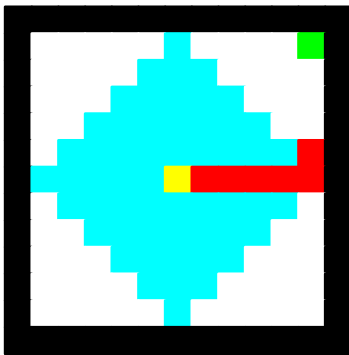


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

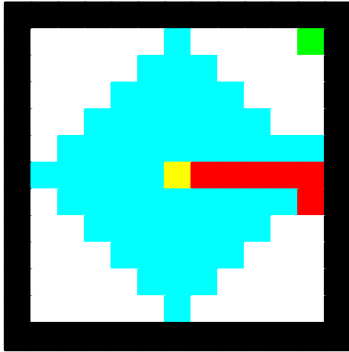


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

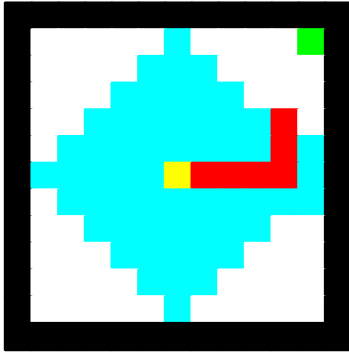


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

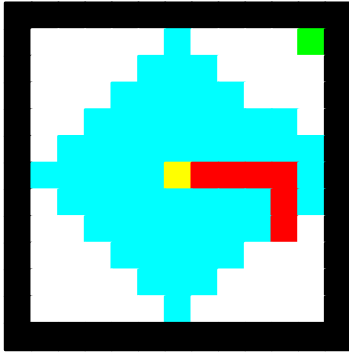


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



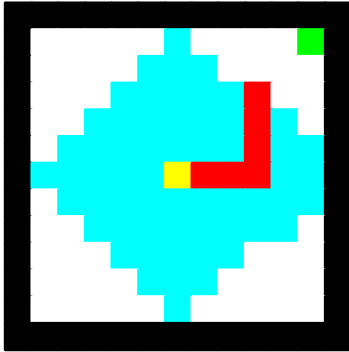


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

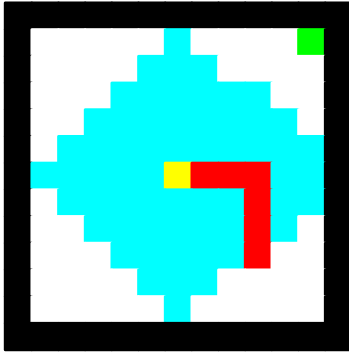


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

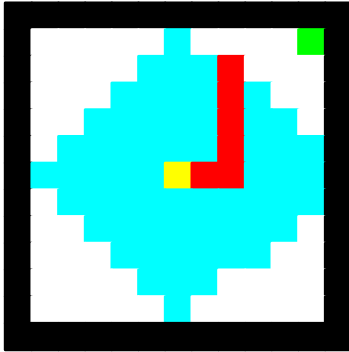


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

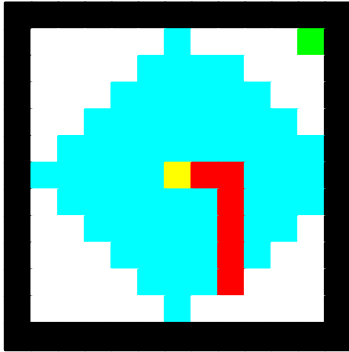


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

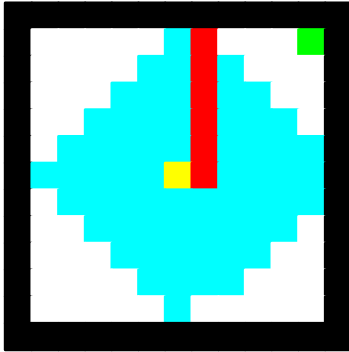


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

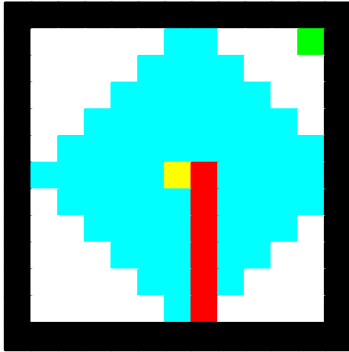


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

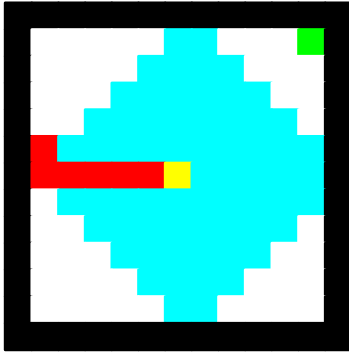


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

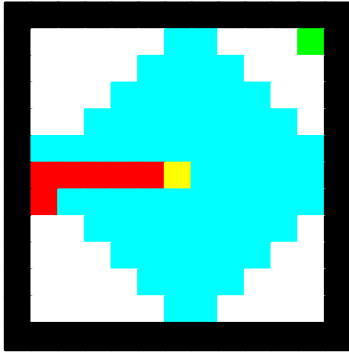


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



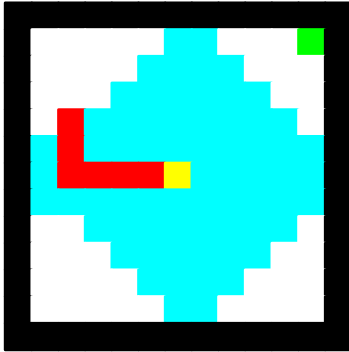


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

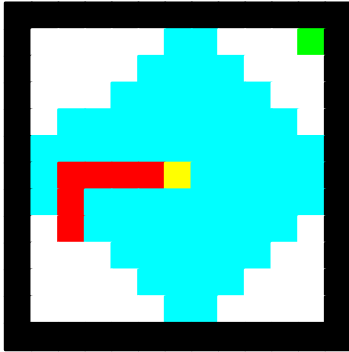


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

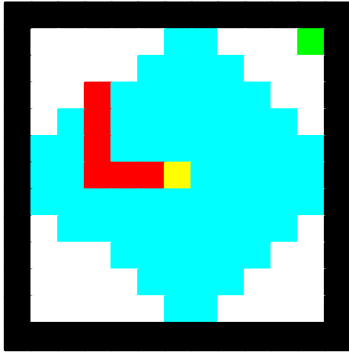


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

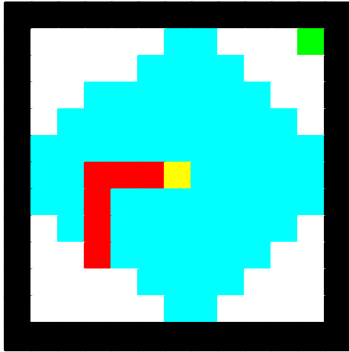


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

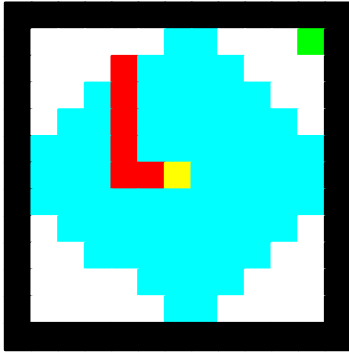


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

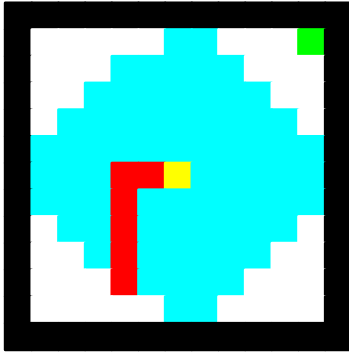


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

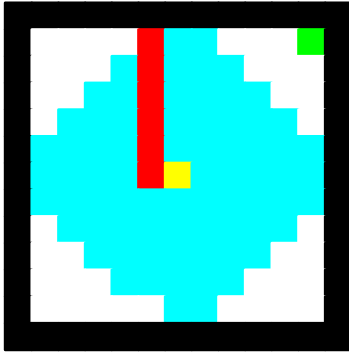


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

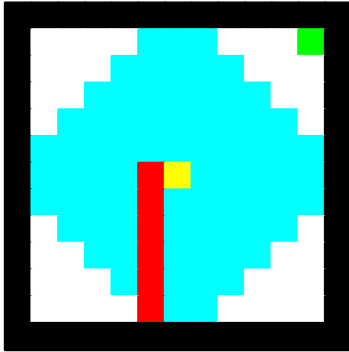


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



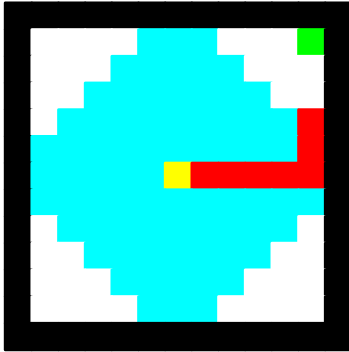


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

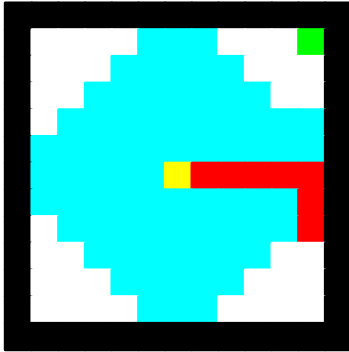


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

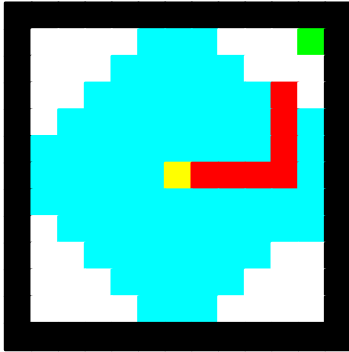


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

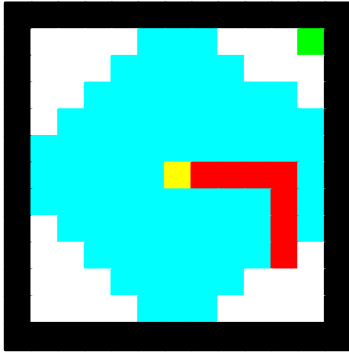


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

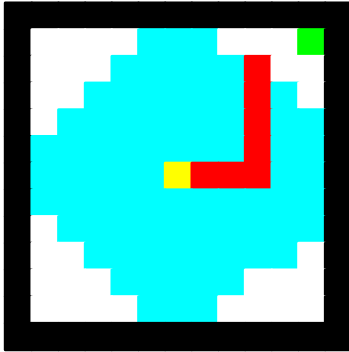


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

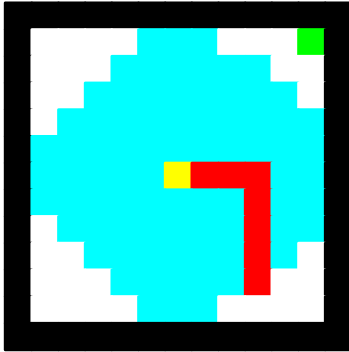


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

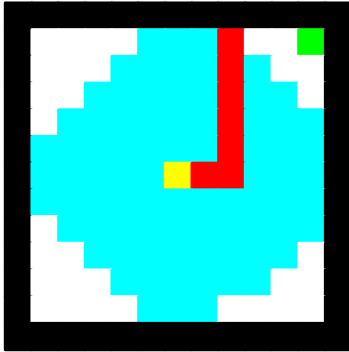


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

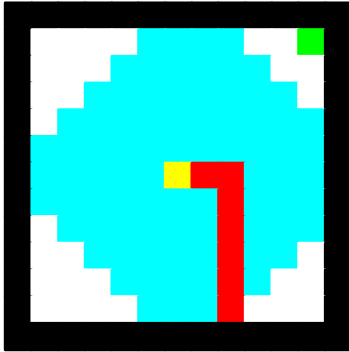


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



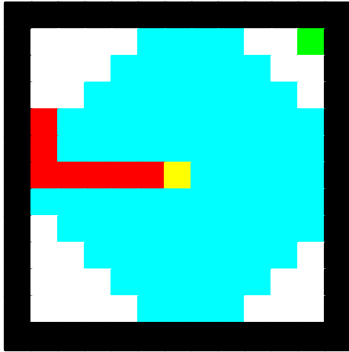


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

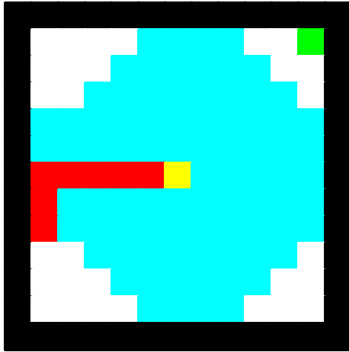


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

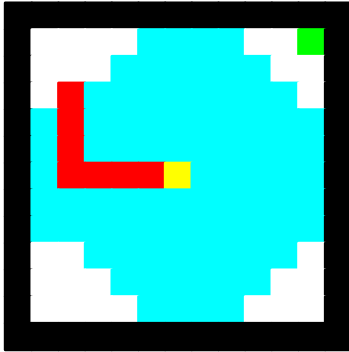


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

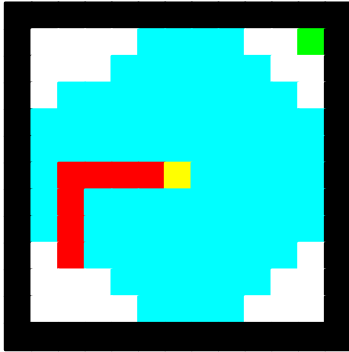


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

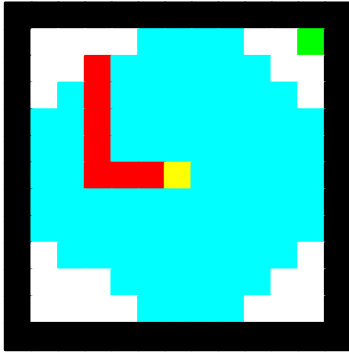


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

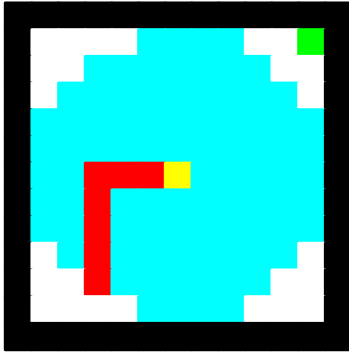


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

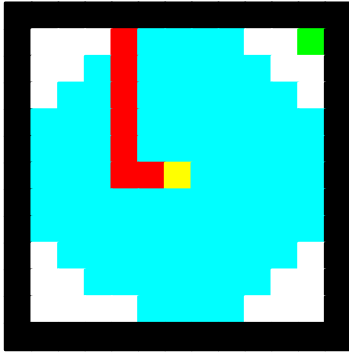


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

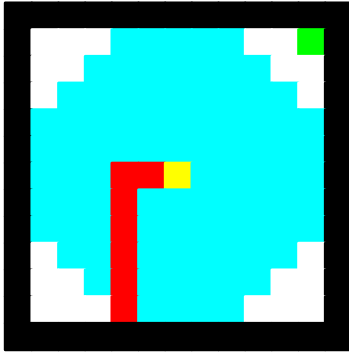


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



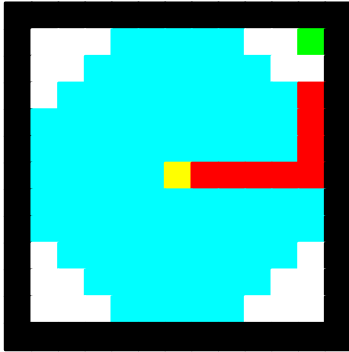


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

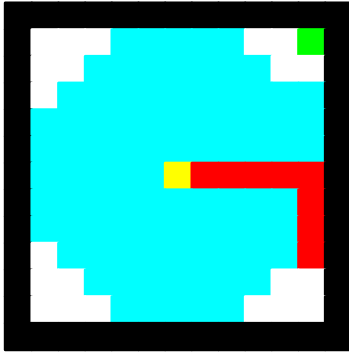


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

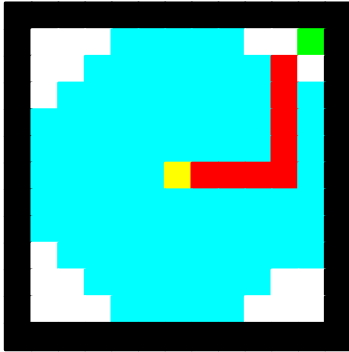


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

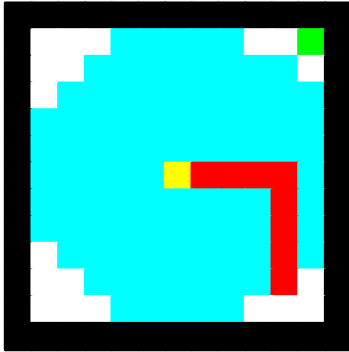


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

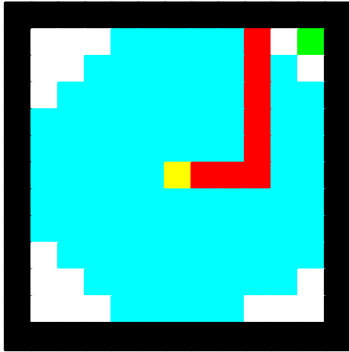


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

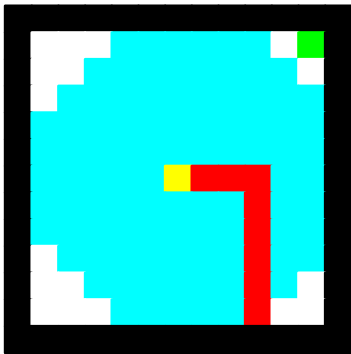


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

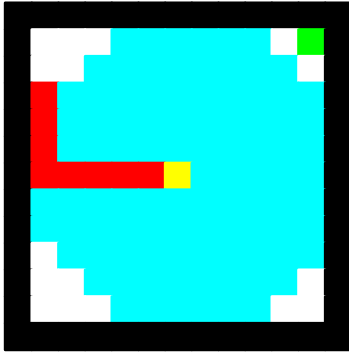


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

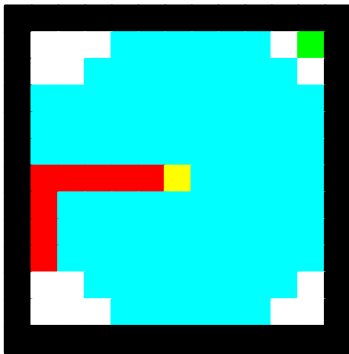


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



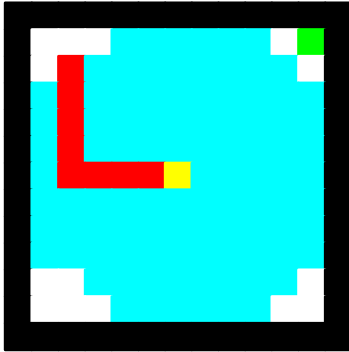


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

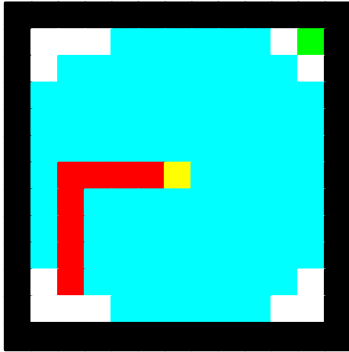


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

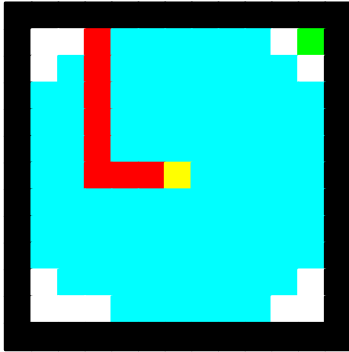


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

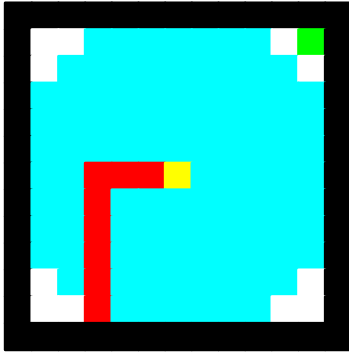


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

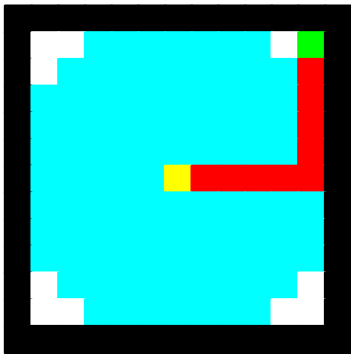


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

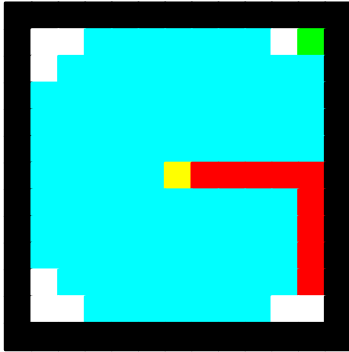


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

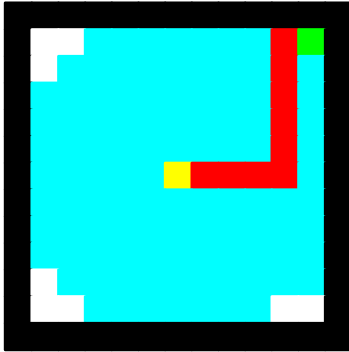


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

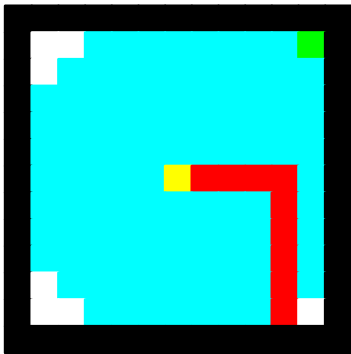


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



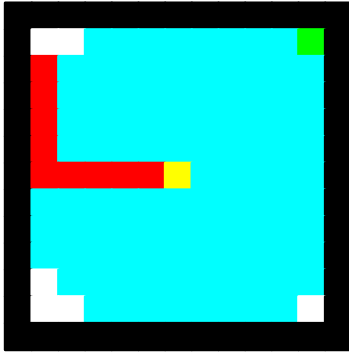


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

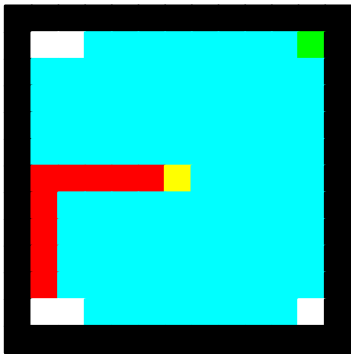


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

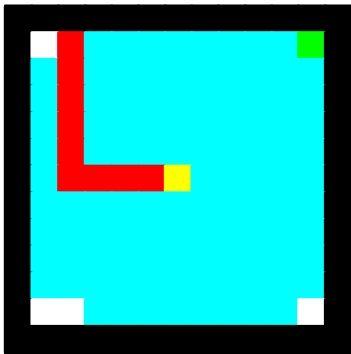


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

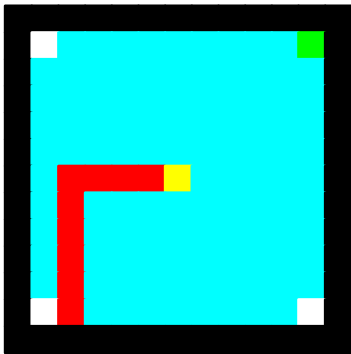


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.

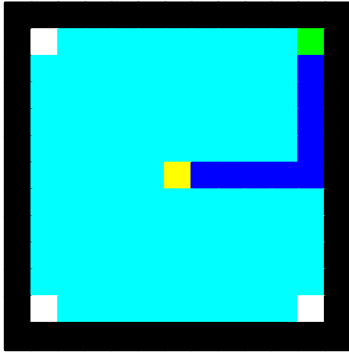


## Problem?

---

So far, our searches have radiated outward from the starting point.

We only notice the goal when we stumble upon it. Wasting time searching on the **wrong side of the goal**.



## Heuristics

---

So far, we have sorted our agenda based on:

$g(n)$  = the path cost from the start to  $n$

Thus, any values with the same  $g(n)$  will be considered at roughly the same time, regardless of whether they are moving "in the right direction" or not.

We can do better by including another function, which we'll call a **heuristic**:

$h(n)$  = an estimate of the remaining cost from  $n$  to the goal

Since we want to minimize total path cost, we can instead sort by:

$f(n) = g(n) + h(n)$

$f(n)$  = an estimate of total cost from start to goal through  $n$

## Heuristics

---

Including a heuristic function makes this an "informed" search: it uses information not only about the path so far, but about the nature of the goal, to focus its attention toward the goal. The resulting algorithm (UC search with a heuristic) is often referred to as  $A^*$  ("A-star").

This can have a dramatic effect on the amount of the search space we explore.

## Heuristics and Optimality

---

In order to guarantee that we still get an optimal path, our heuristic needs a couple of properties:

- **admissibility:**  $h(n)$  never overestimates the actual cost of the lowest-cost path from  $n$  to the goal
- **consistency:** roughly,  $h(n)$  does not increase as  $n$  gets closer to the goal.

The ideal heuristic should be

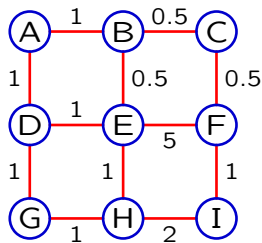
- as close as possible to actual cost (without exceeding it)
- easy to calculate



## Example

---

Find path  $E \rightarrow I$ , A\*, heuristic:  $h(s) = M(s, I)/2$



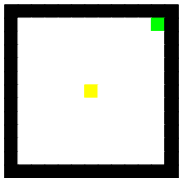
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



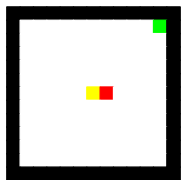
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



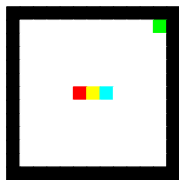
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



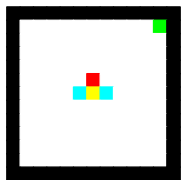
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



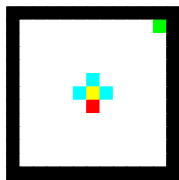
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



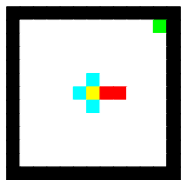
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



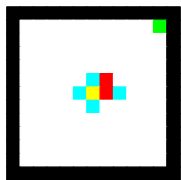
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





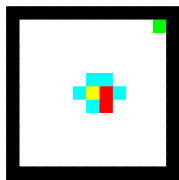
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



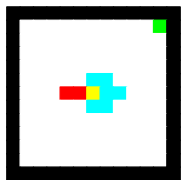
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



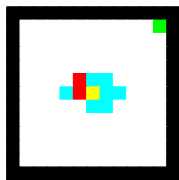
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



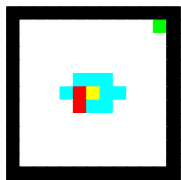
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



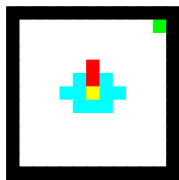
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



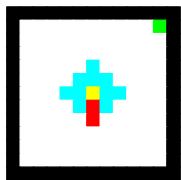
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



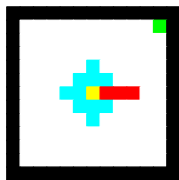
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



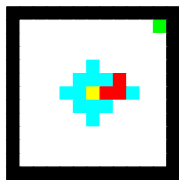
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





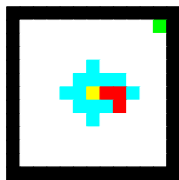
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



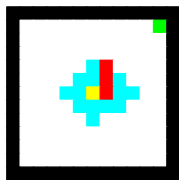
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



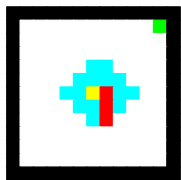
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



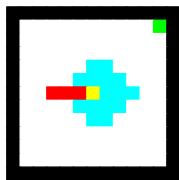
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



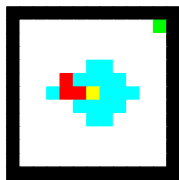
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



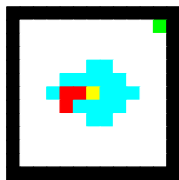
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



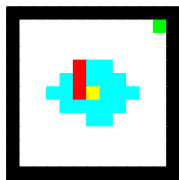
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



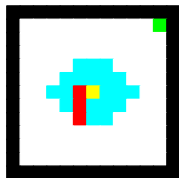
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





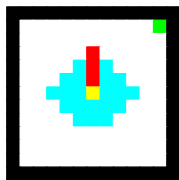
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



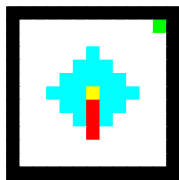
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



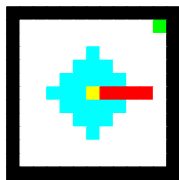
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



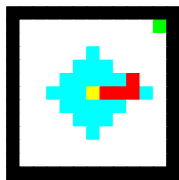
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



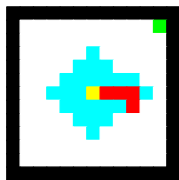
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



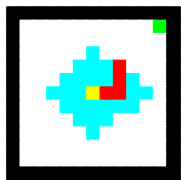
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



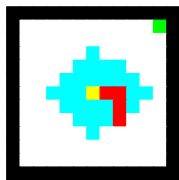
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



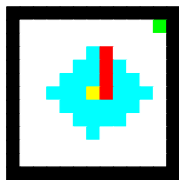
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





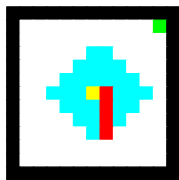
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



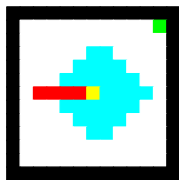
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



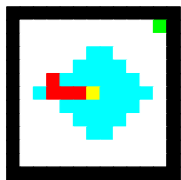
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



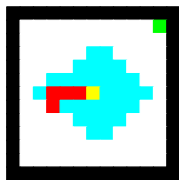
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



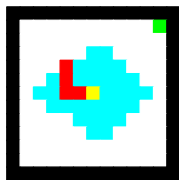
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



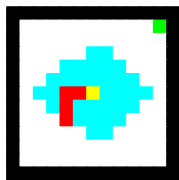
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



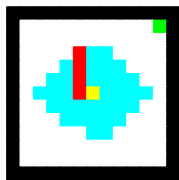
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



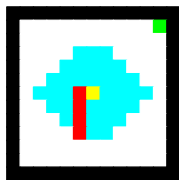
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





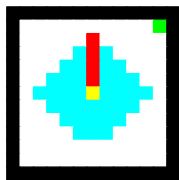
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



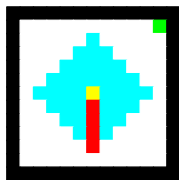
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



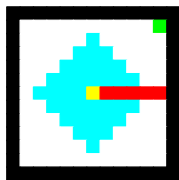
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



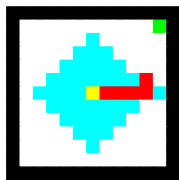
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



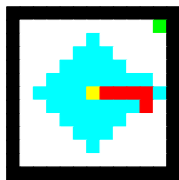
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



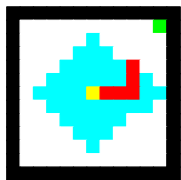
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



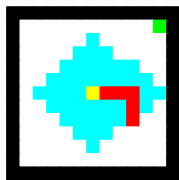
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



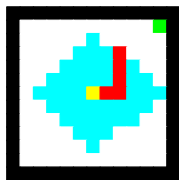
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





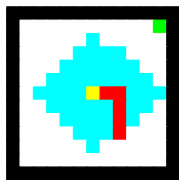
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



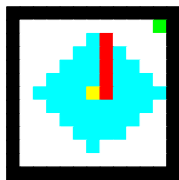
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



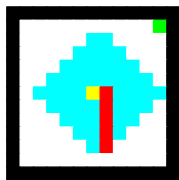
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



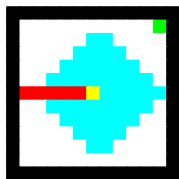
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



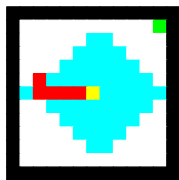
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



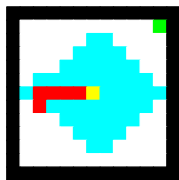
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



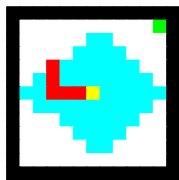
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



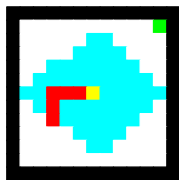
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





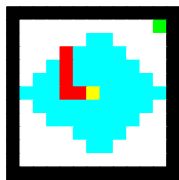
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



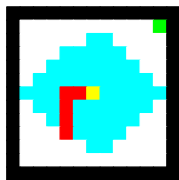
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



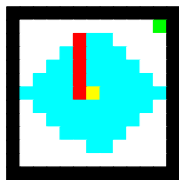
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



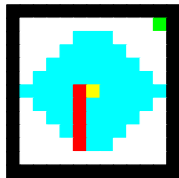
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



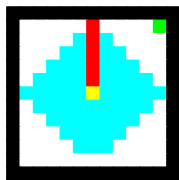
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



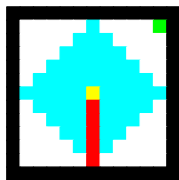
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



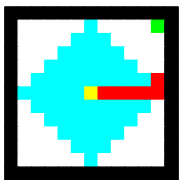
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



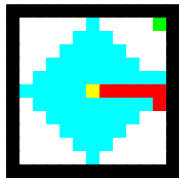
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





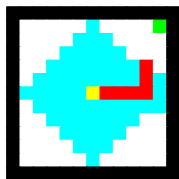
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



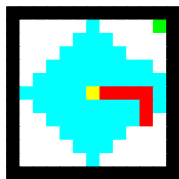
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



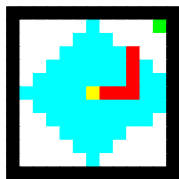
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



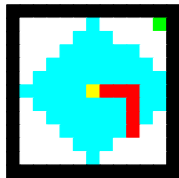
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



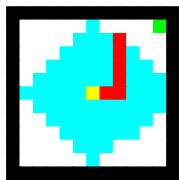
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



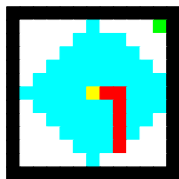
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



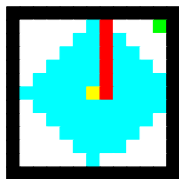
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



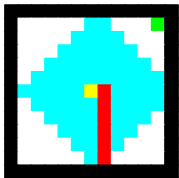
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





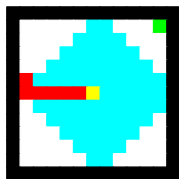
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



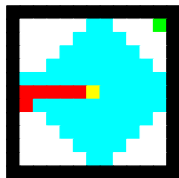
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



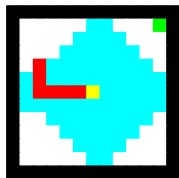
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



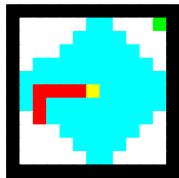
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



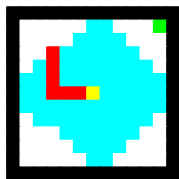
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



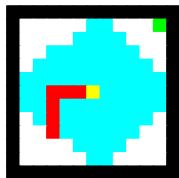
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



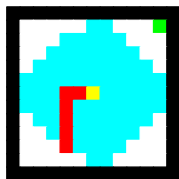
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





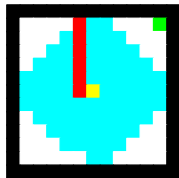
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



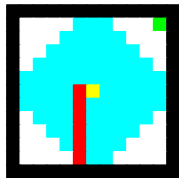
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



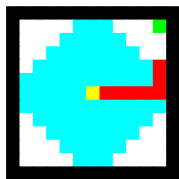
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



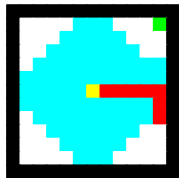
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



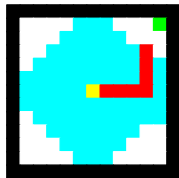
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



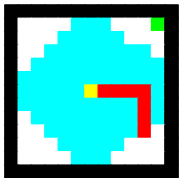
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



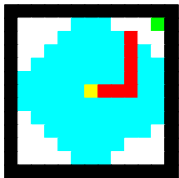
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



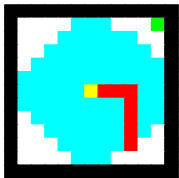
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





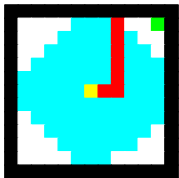
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



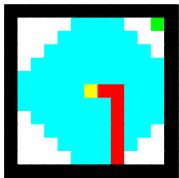
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



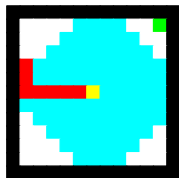
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



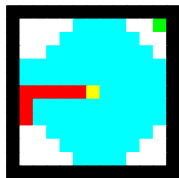
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



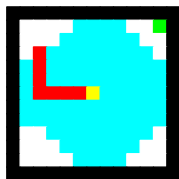
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



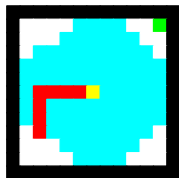
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



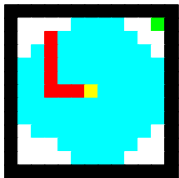
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



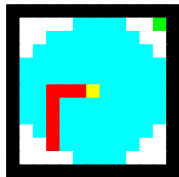
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



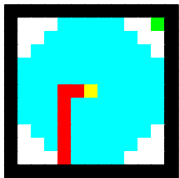
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



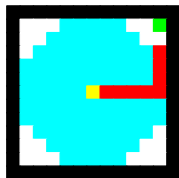
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



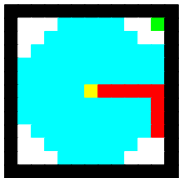
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



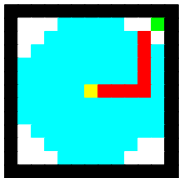
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



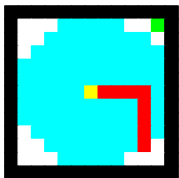
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



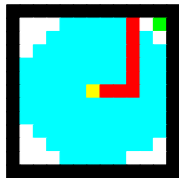
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



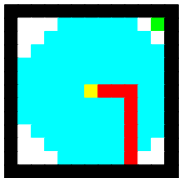
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





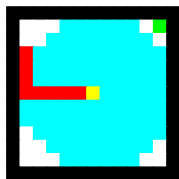
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



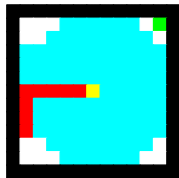
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



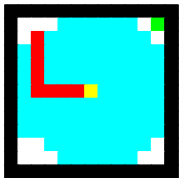
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



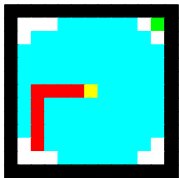
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



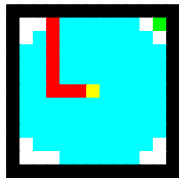
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



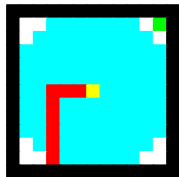
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



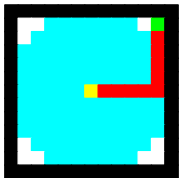
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



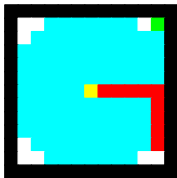
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$





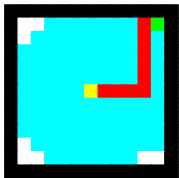
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



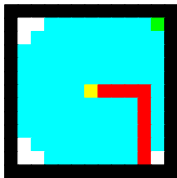
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



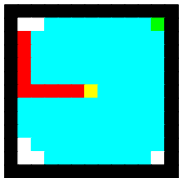
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



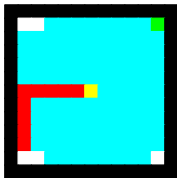
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



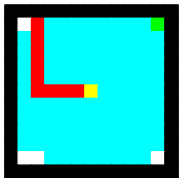
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



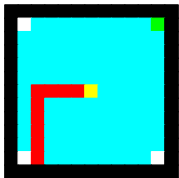
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



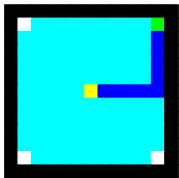
## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

How many of the following heuristics are **admissible**?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



## Check Yourself!

---

Consider searching in a 4-direction grid, and let  $(r_0, c_0)$  and  $(r_1, c_1)$  represent the current and goal locations.

Which of the admissible heuristics minimizes the number of paths expanded?

1.  $\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$  (i.e. Manhattan distance)
2.  $\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
3.  $\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
4.  $2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$
5.  $2 * \max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$



## Compare

---

Heuristic	Paths Expanded
$\text{abs}(r_0 - r_1) + \text{abs}(c_0 - c_1)$	42
$\min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$	114
$\max(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$	60
$2 * \min(\text{abs}(r_0 - r_1), \text{abs}(c_0 - c_1))$	72

## Heuristics: Eight Puzzle

---

Consider three heuristic functions for the "eight puzzle":

- a.  $h(n) = 0$
- b.  $h(n) = \text{number of tiles out of place}$
- c.  $h(n) = \text{sum over tiles of Manhattan distances to their goals}$

Let  $M_i = \#$  of moves in the best solution with heuristic  $i$

Let  $E_i = \#$  of paths expanded during search with heuristic  $i$

Which of the following is/are true?

- 1.  $M_a = M_b = M_c$
- 2.  $E_a = E_b = E_c$
- 3.  $M_a > M_b > M_c$
- 4.  $E_a \geq E_b \geq E_c$

## Compare

---

Heuristic	Expanded Solution	Cost
-----------	-------------------	------

a	84,516	22
---	--------	----

b	8,329	22
---	-------	----

c	1,348	22
---	-------	----

**End**

---

**End**

---