CSCI 132:
Basic Data Structures and Algorithms

Program 5 Discussion

Reese Pearsall

Spring 2025

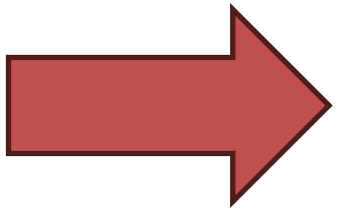
Announcements

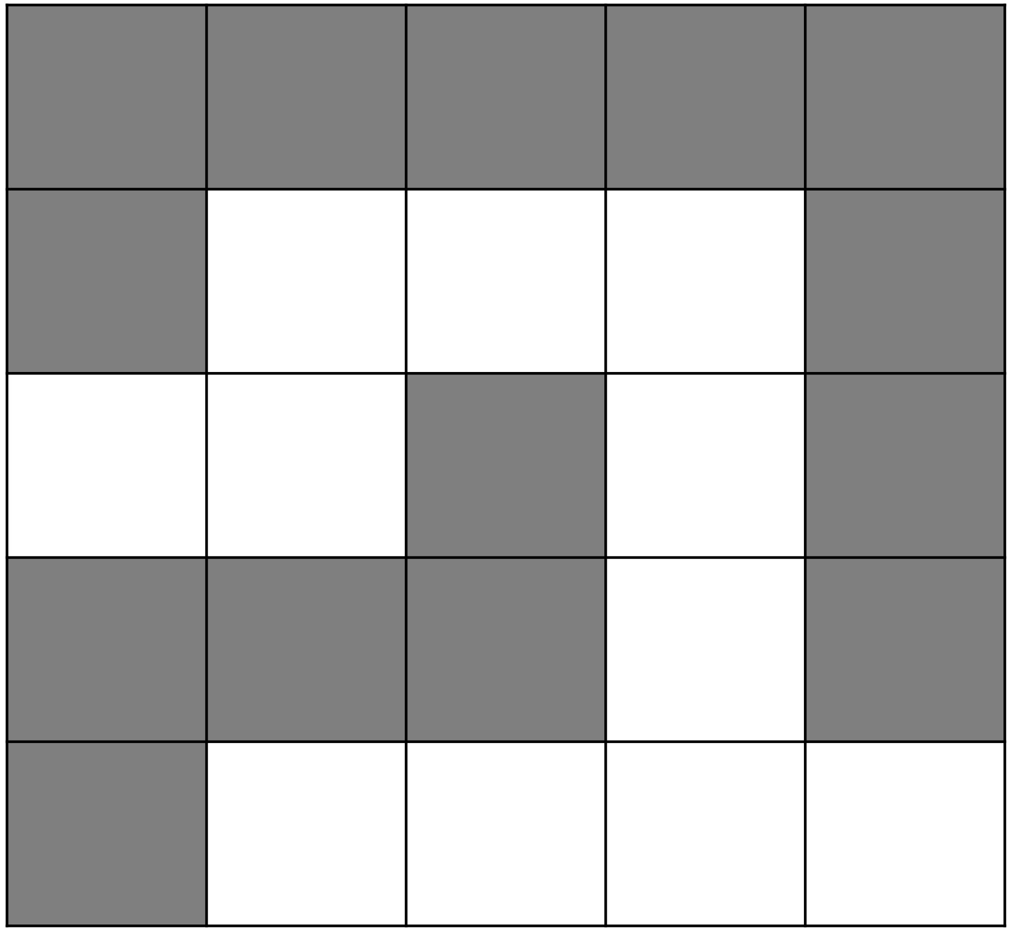
- Program 5 posted, Sunday due 5/4

- Rubber duck extra credit posted. Due next Friday

```
char[][] maze
```

```
[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
 ]
```

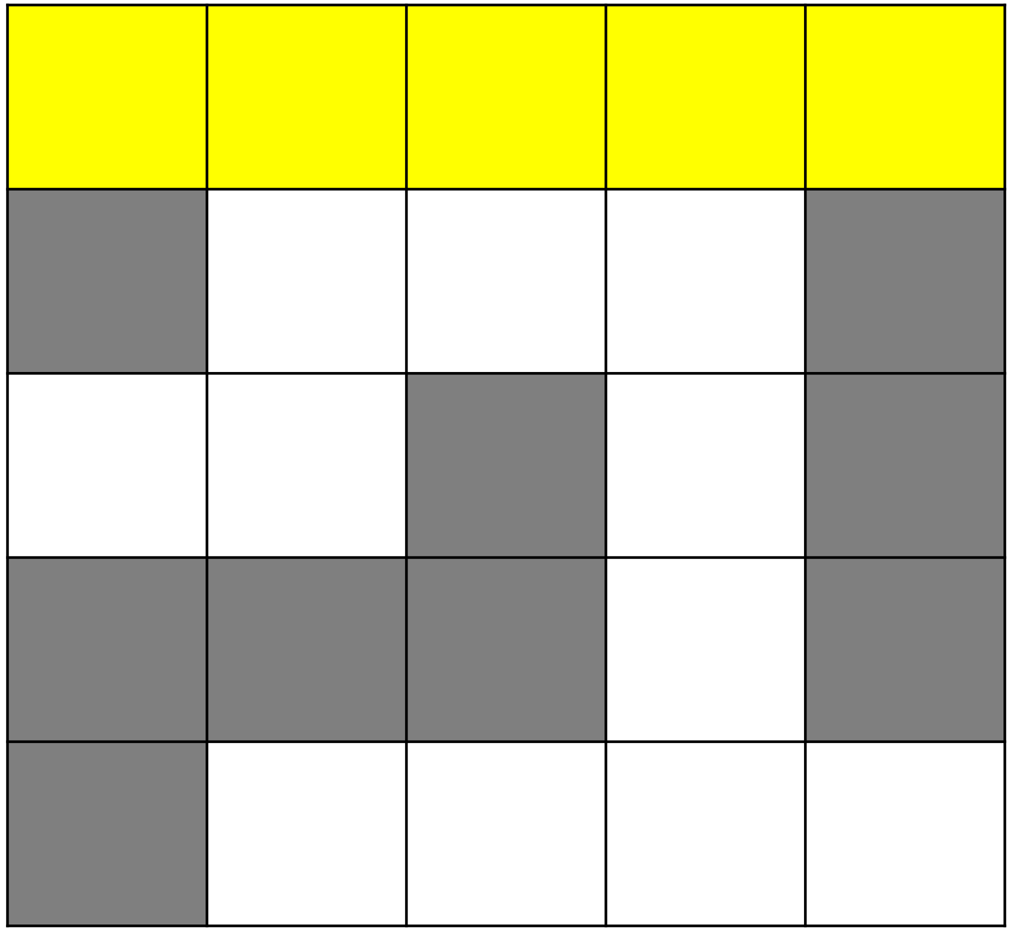




```
char[][] maze
```

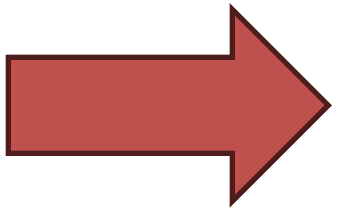
```
[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
 ]
```

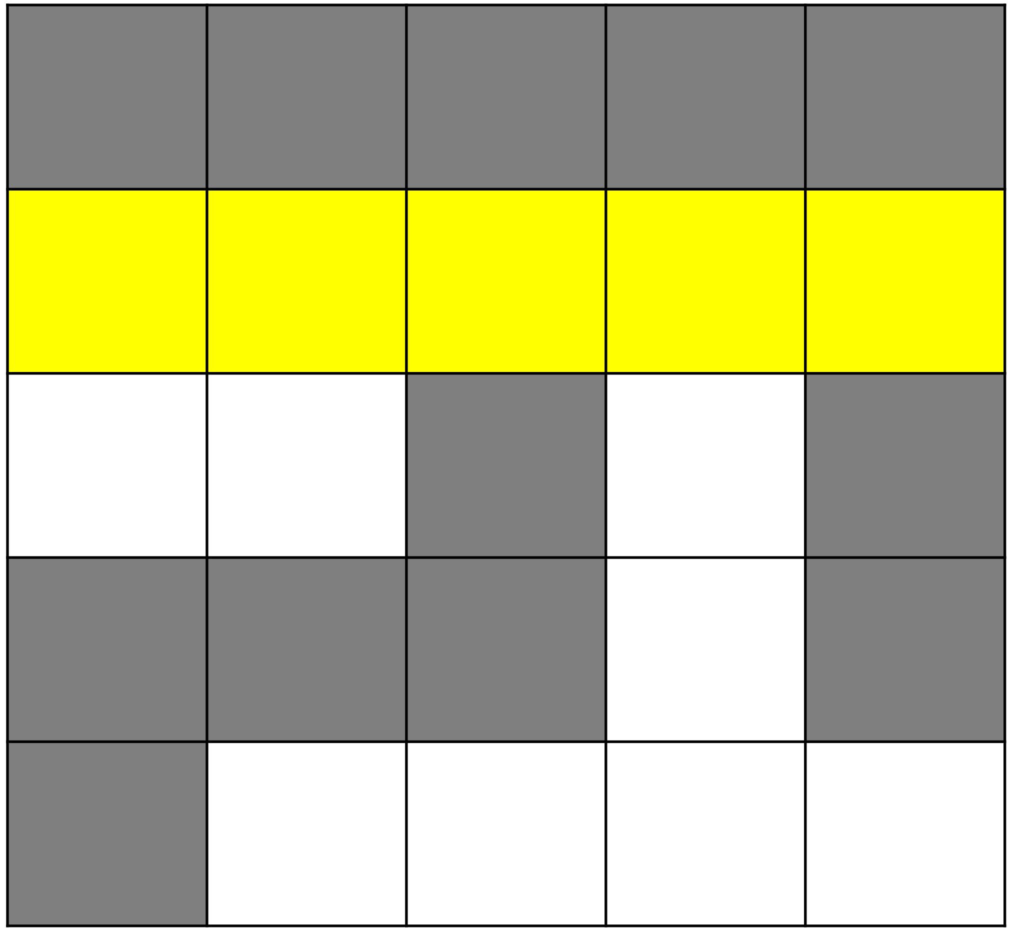
```
maze[0]
```



`char[][] maze`

```
[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
]
```



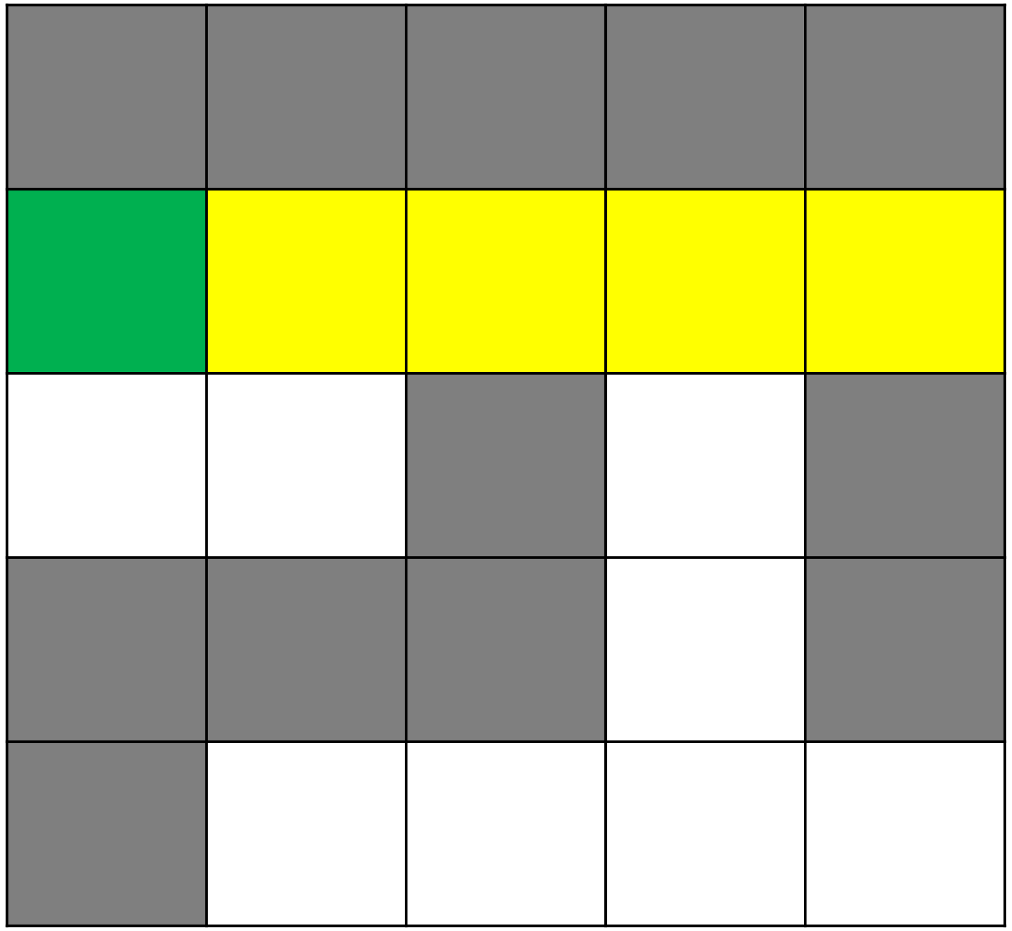


`maze[1]`

```
char[][] maze
```

```
[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
]
```

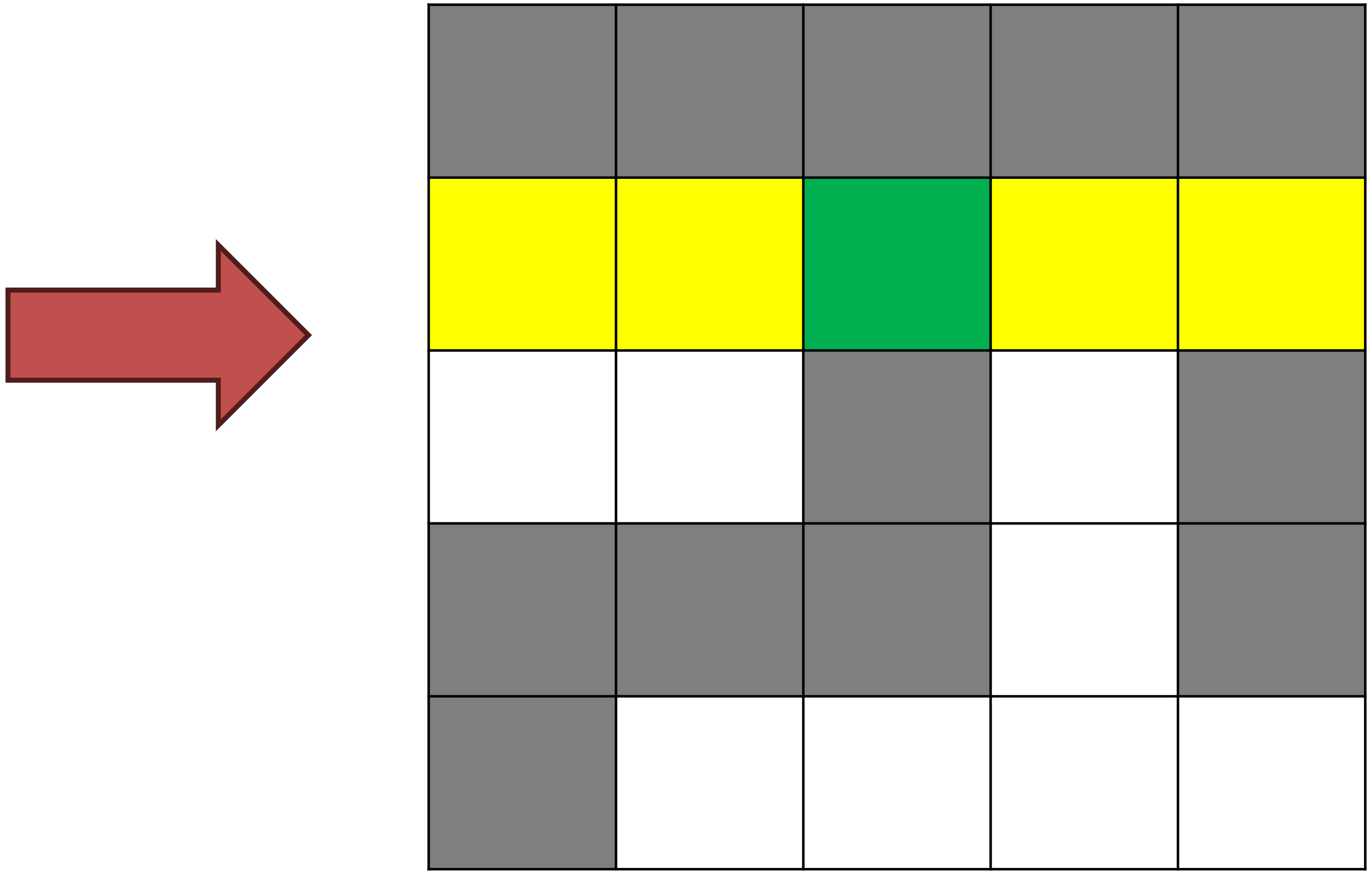
```
maze[1][0]
```



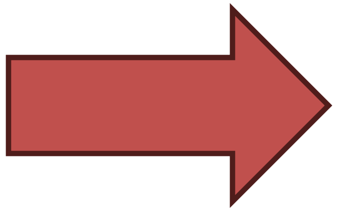
`char[][] maze`

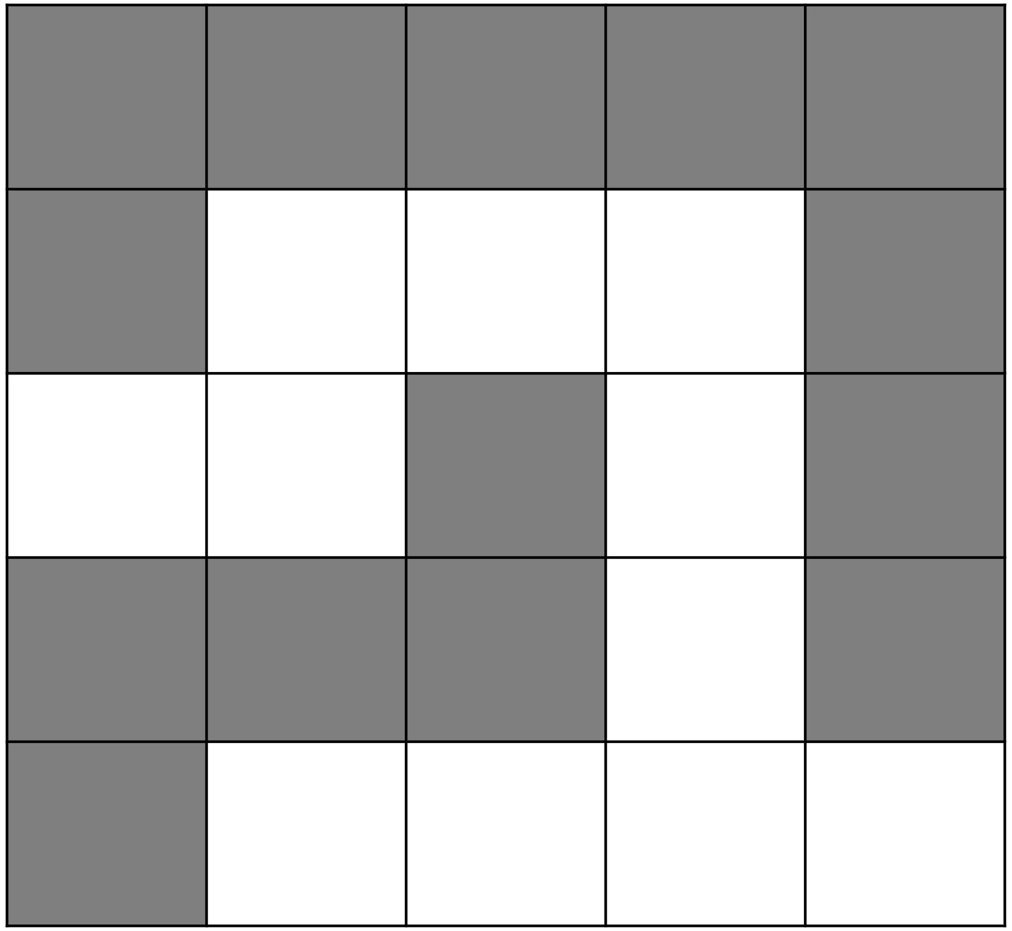
```
[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
 ]
```

`maze[1][2]`

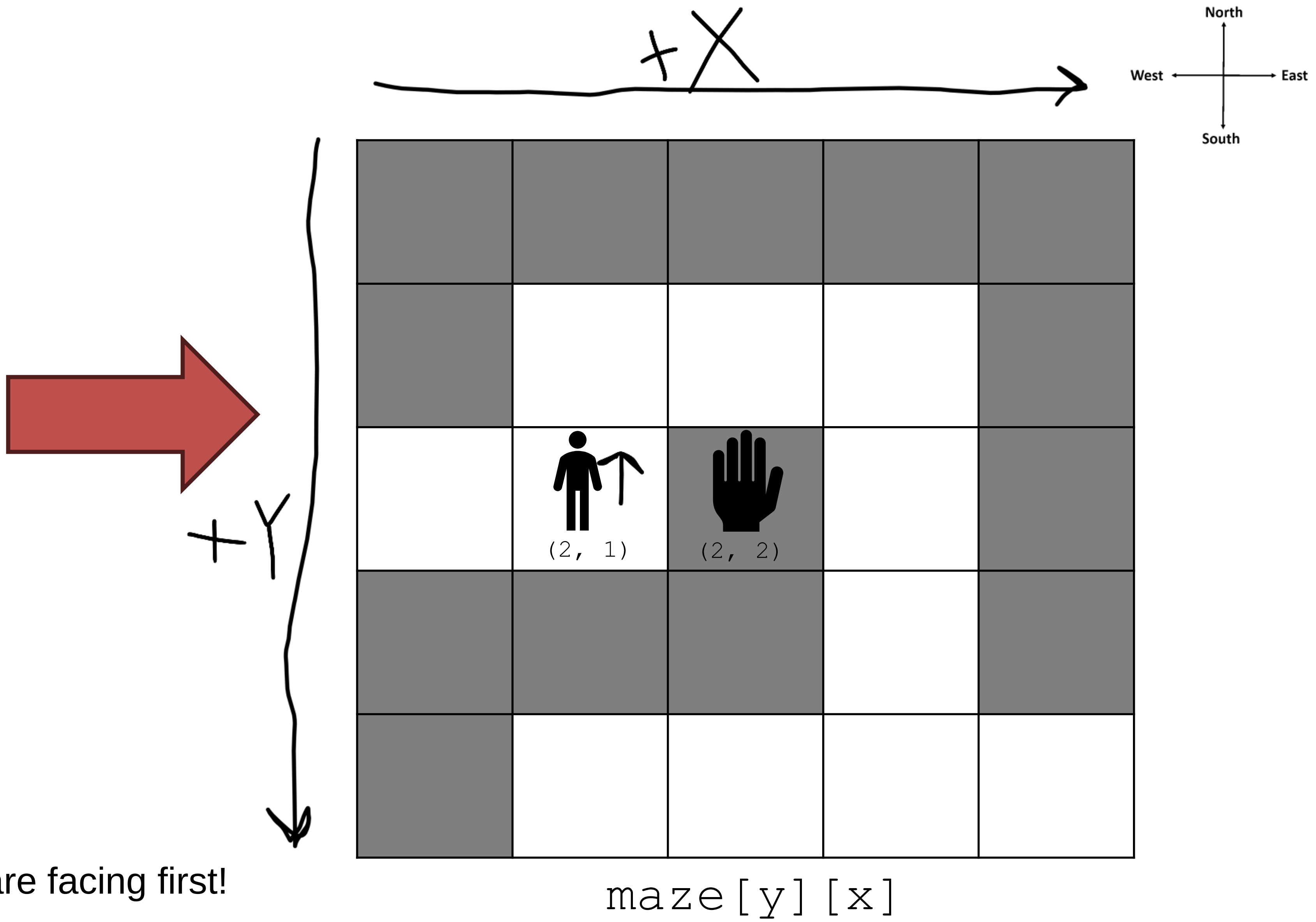


```
char[][] maze

[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
 ]

maze[y][x]
```





```
char[][] maze

[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
 ]
```



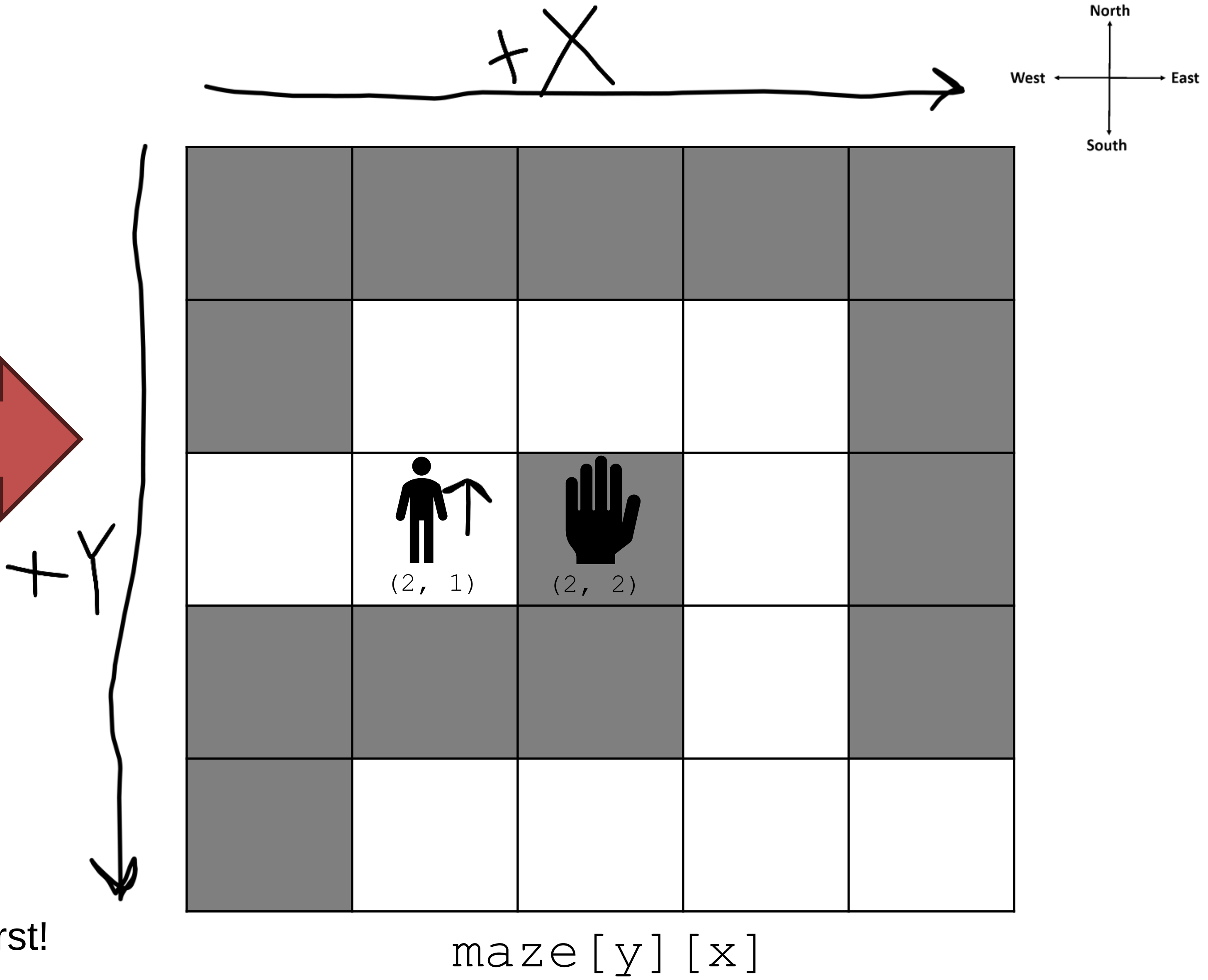
Goal: Move forward one spot

We need to know which direction we are facing first!

How do we know direction we are facing?

```
char[][] maze
```

```
[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
 ]
```



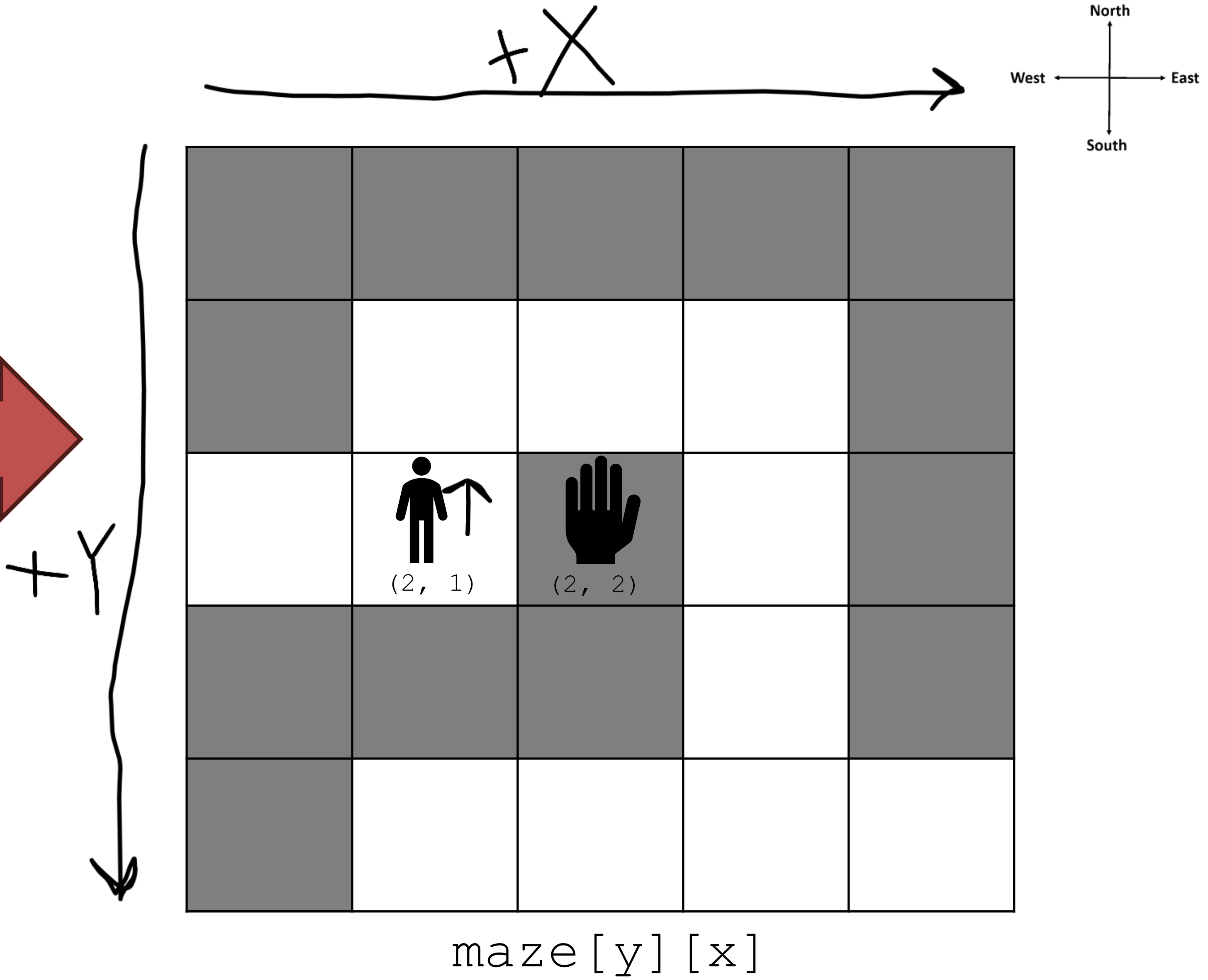
Goal: Move forward one spot

We need to know which direction we are facing first!

Our character Y value and our hand's Y value is the same,
And our character's X value is *less than* our hands' X value

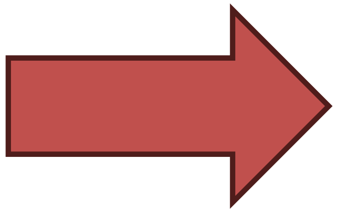
```
      char[][] maze

[ [ #, #, #, # ,#],
  [ #, . , . , . , #],
  [ . , . ,# , . , #],
  [ #, #, #, . , #],
  [ #, . , . , . , . ],
 ]
```

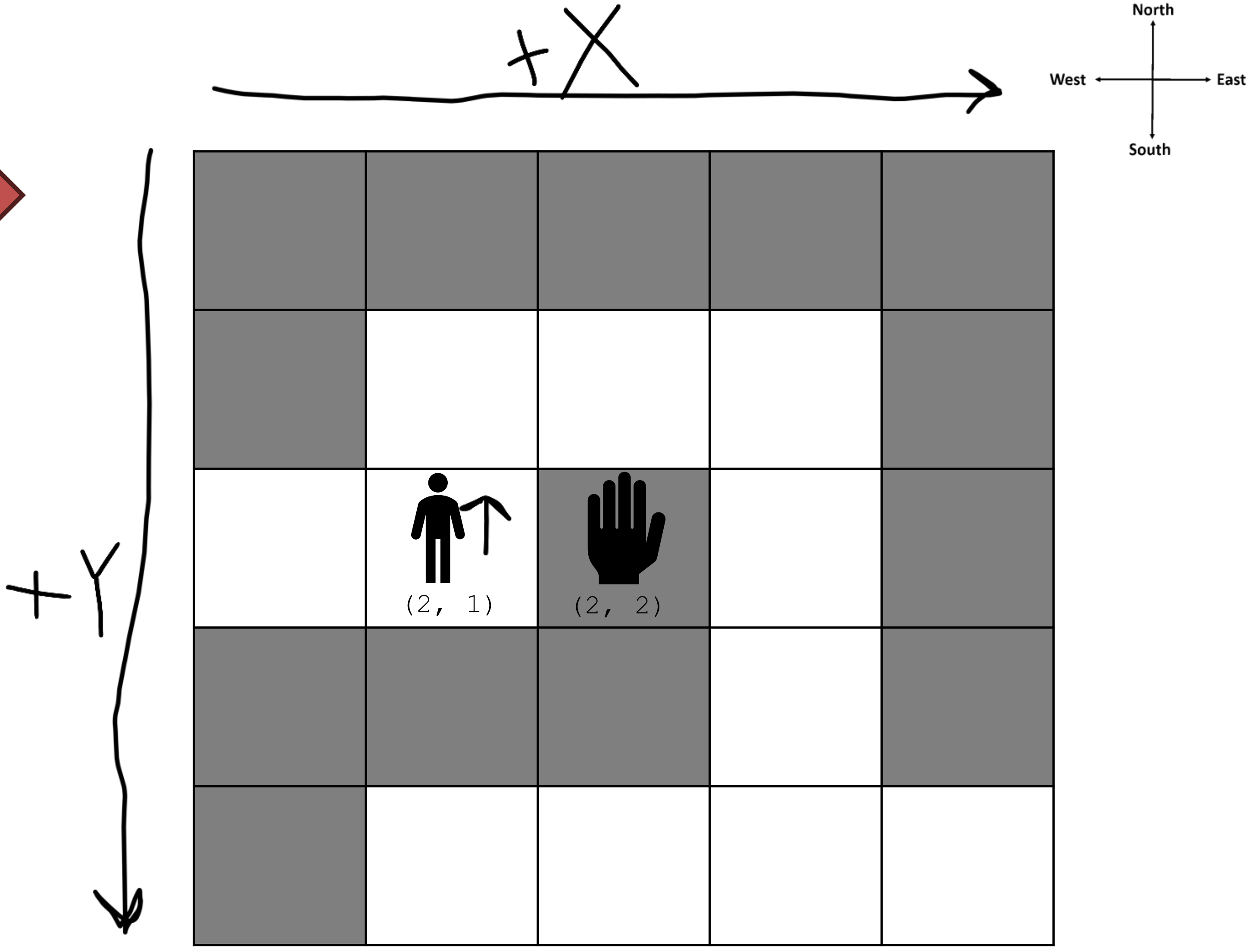
```
if(y == hand_y && hand_x > x)
        direction = "North";
}
```



char[][] maze

```
[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , .],
]
```





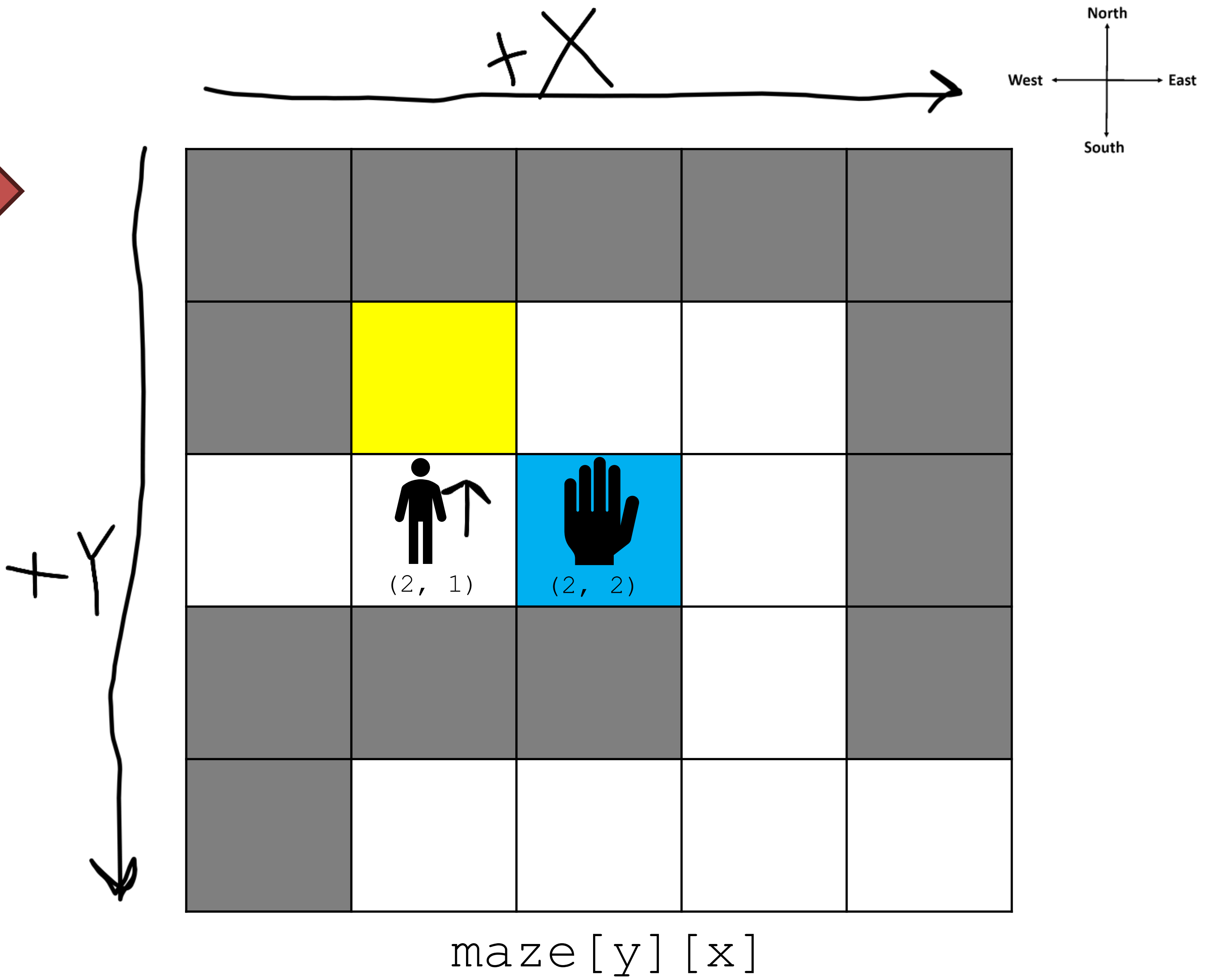
maze[y][x]

```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
```

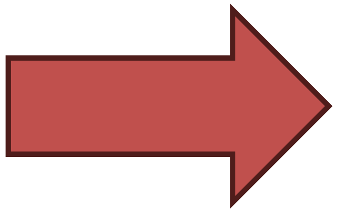
How do we detect if we can move forward?

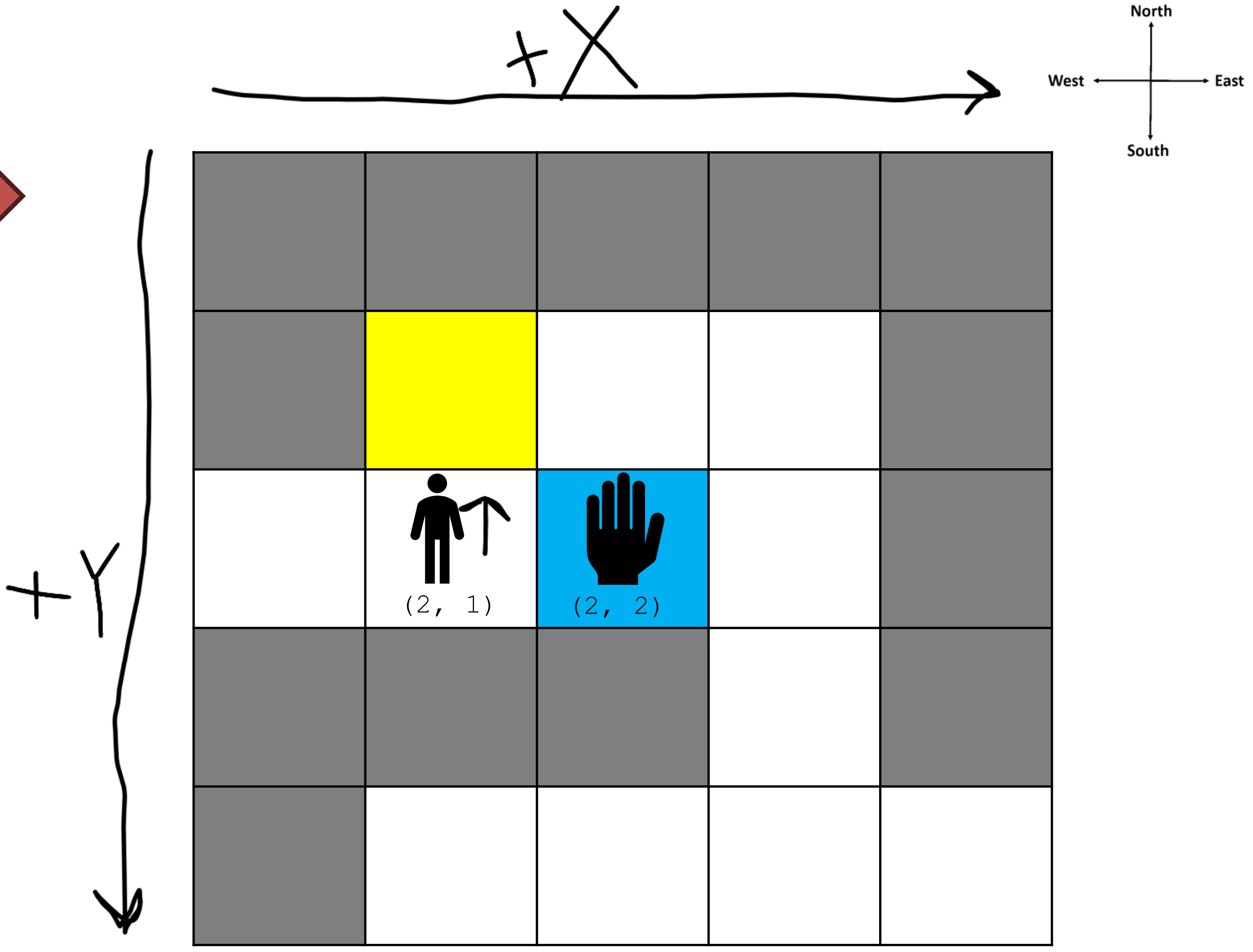
```
        char[][] maze

    [[#, #, #, # ,#],
     [#, ., ., ., #],
     [., ., #, ., #],
     [#, #, #, ., #],
     [#, ., ., ., .],
    ]
```

```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){


}
```



```
char[][] maze

[[#, #, #, # ,#],
 [#, ., . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , .],
]
```





maze[y][x]

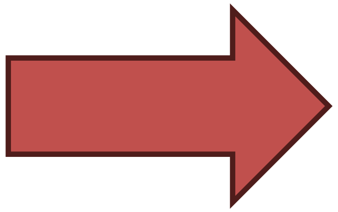
```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){
```

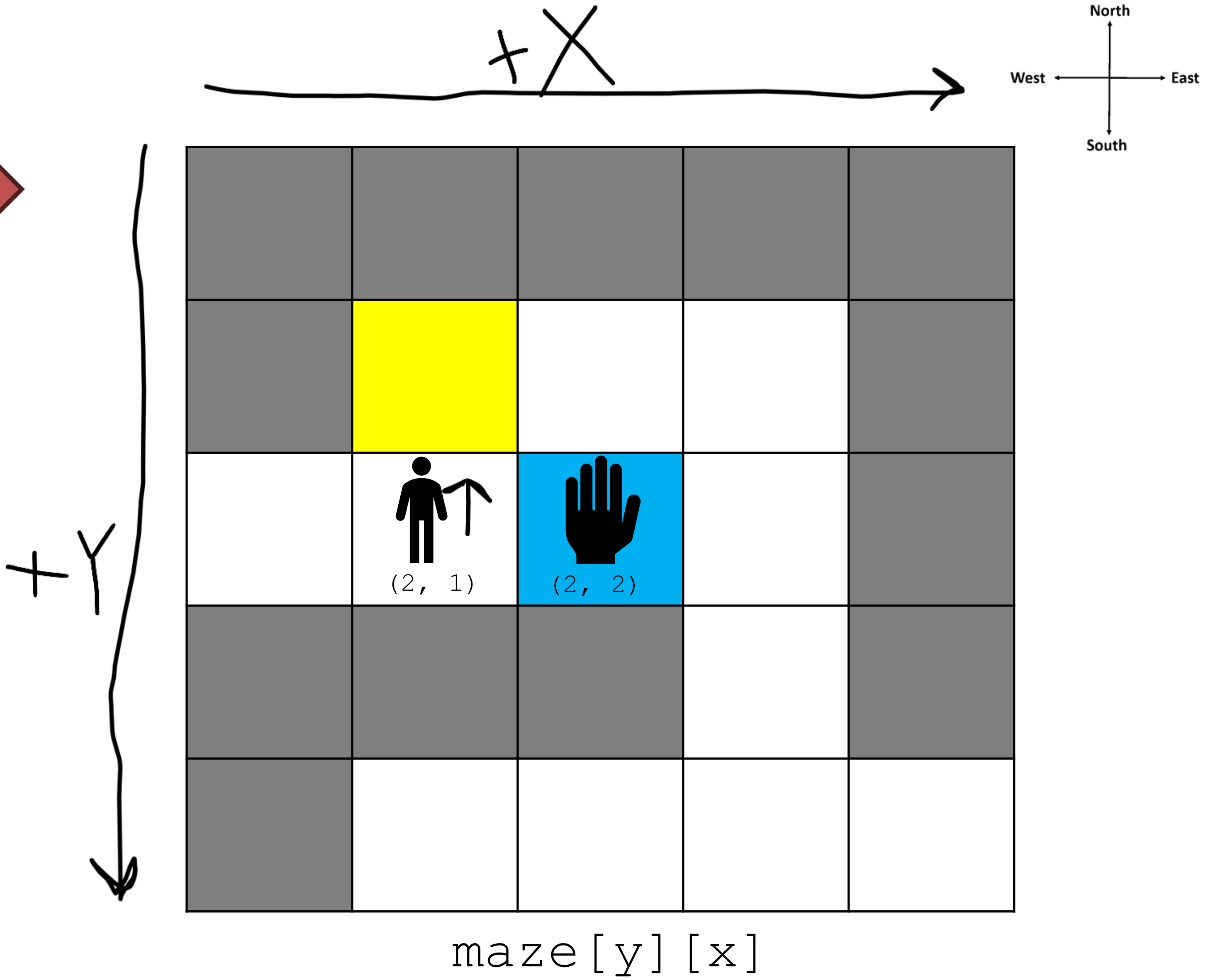
Make one move by recursively calling the method with the new values

```
}
```

`makeMove(x, y, hand_x, hand_y)`

```
char[][] maze

[[#, #, #, # ,#],
 [#, ., . , . , #],
 [ . , . ,#, . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```





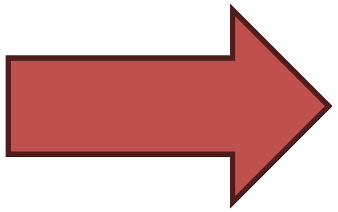
```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){
```

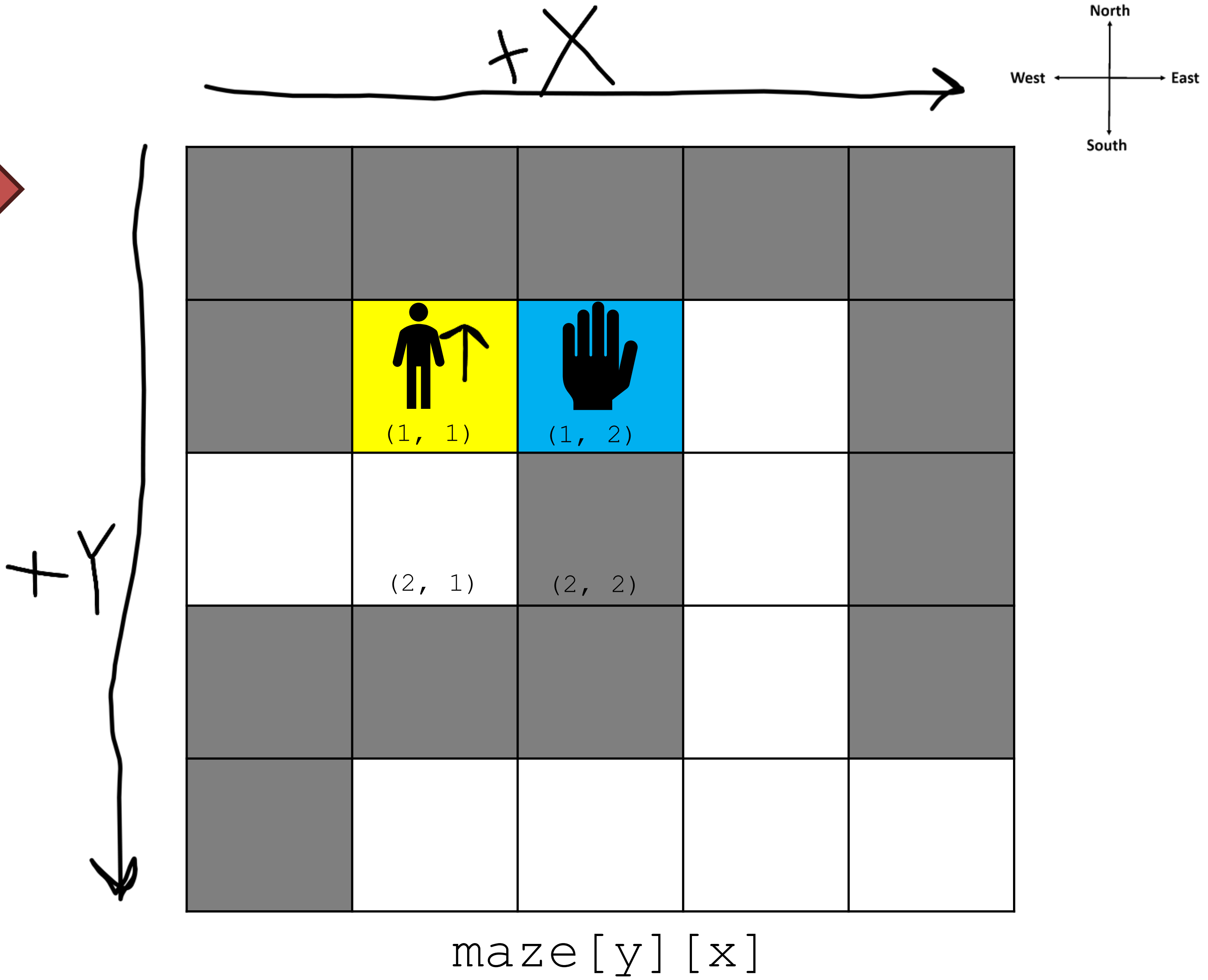
Make one move by recursively calling the method with the new values

```
}
```

```
makeMove(x, y, hand_x, hand_y)
```

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [ . , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```



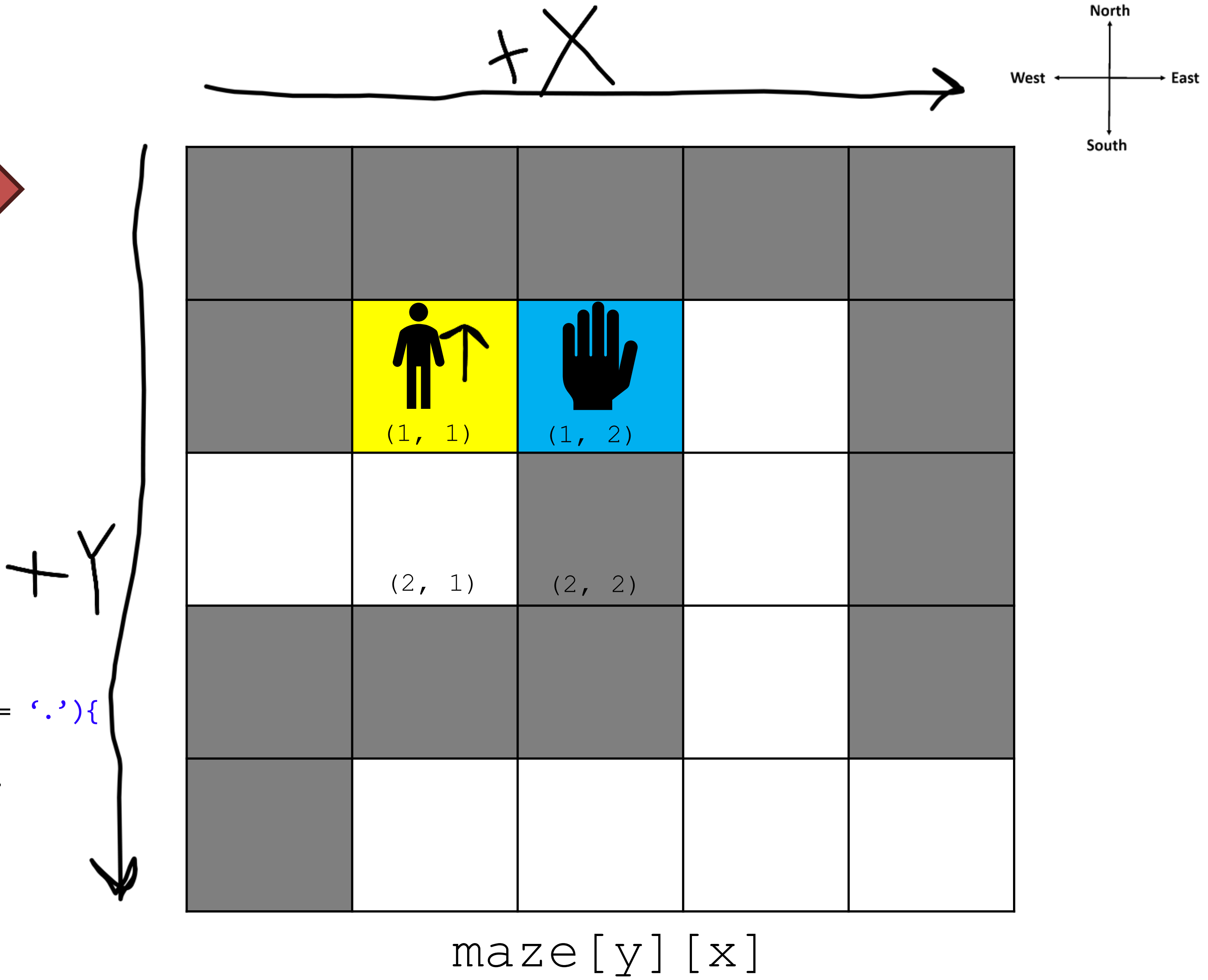


```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

    makeMove(x, y-1, hand_x, hand_y-1);

}
```

```
makeMove(x, y, hand_x, hand_y)
```

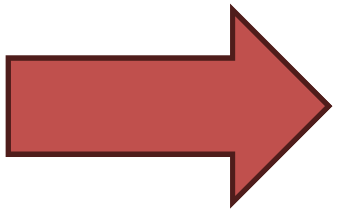
```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , .],
]
```

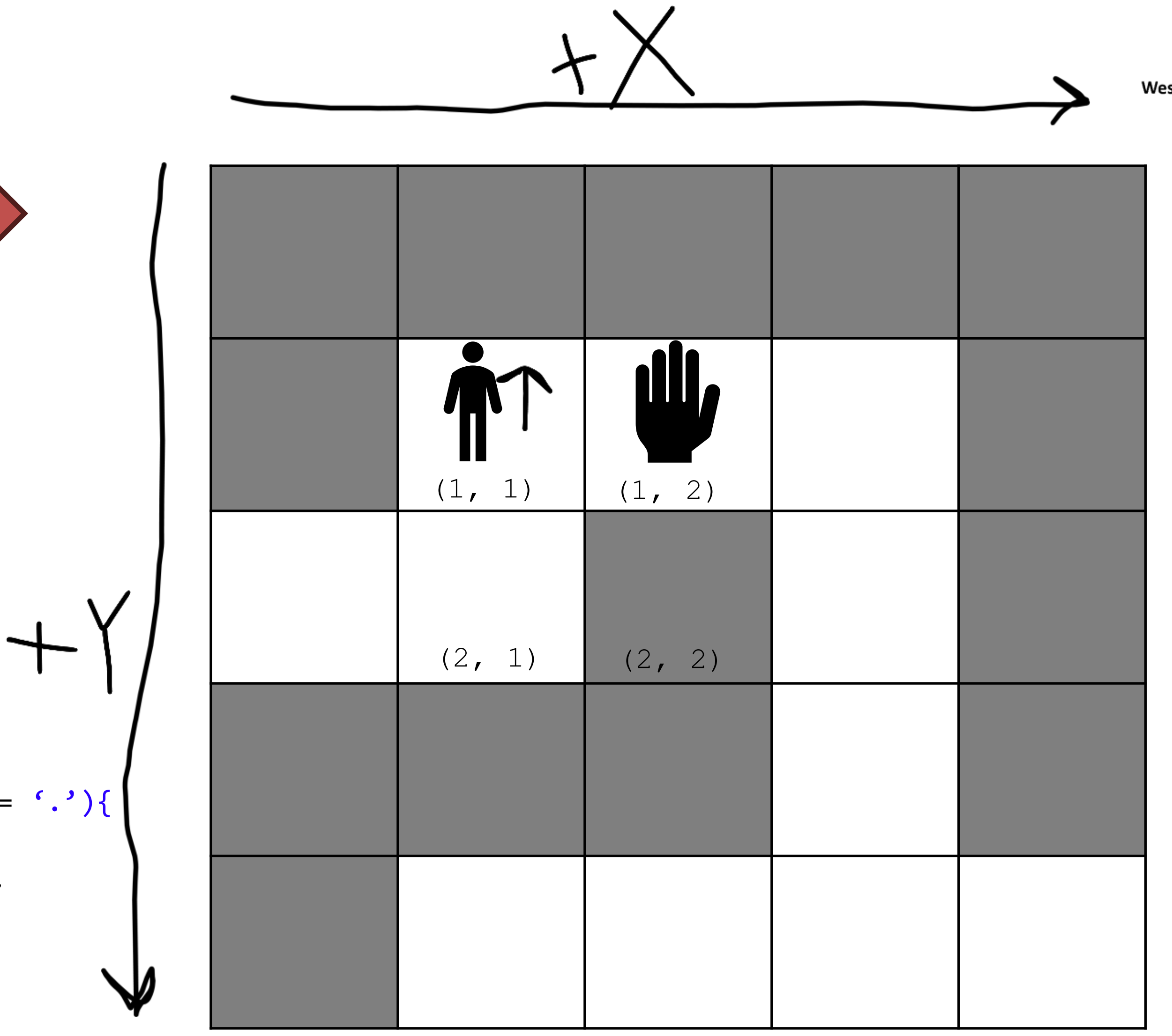
```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
```



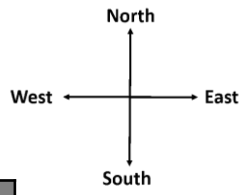
```
makeMove(x, y, hand_x, hand_y)
```

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , .],
]
```





maze[y][x]

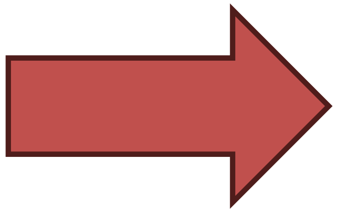


```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
```

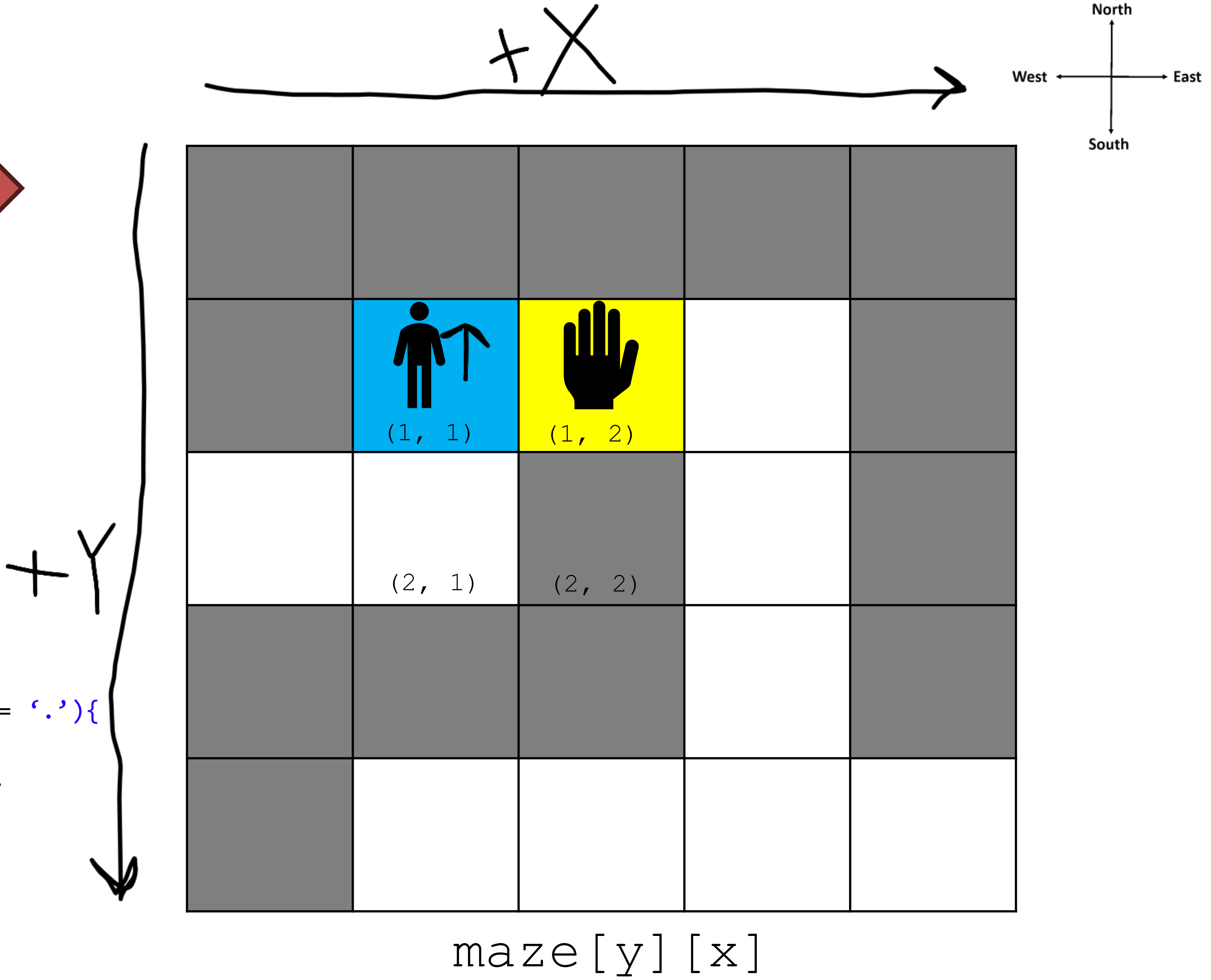
Turn right and move forward one spot?

```
makeMove(x, y, hand_x, hand_y)
```

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```

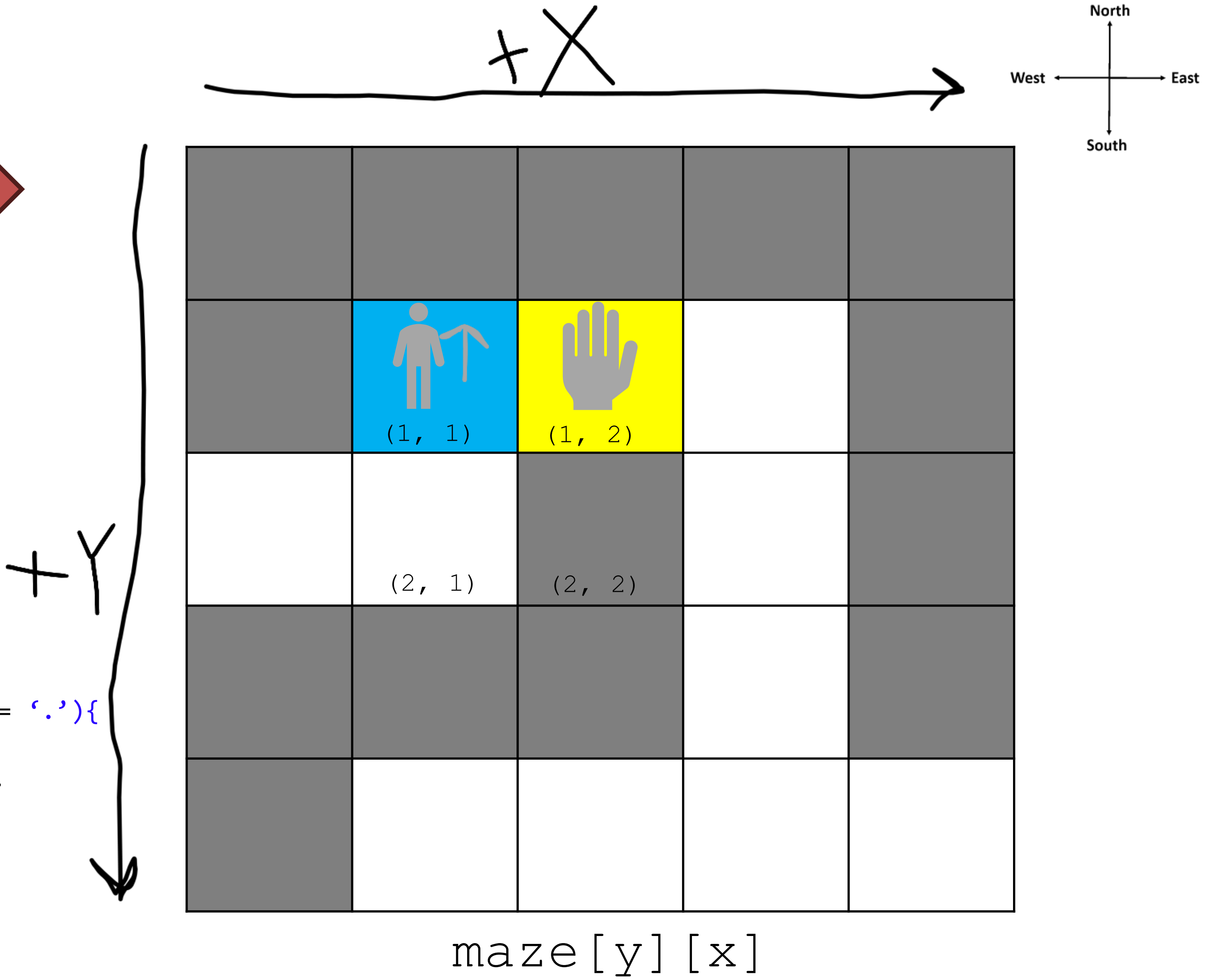


```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
        if(maze[hand_y][hand_x] == '.'){


        }
makeMove(x, y, hand_x, hand_y)
```



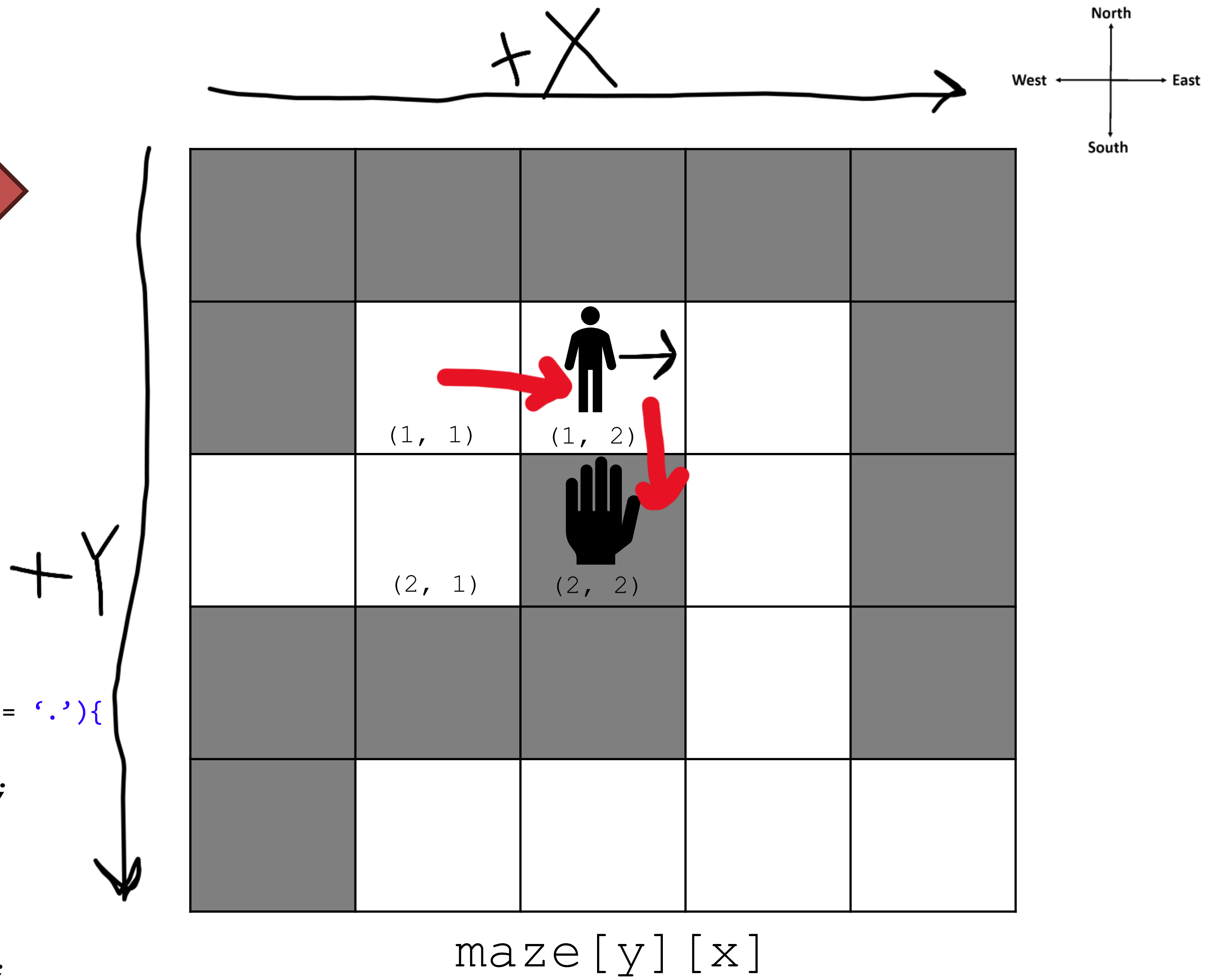
```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [ . , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```

```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
        if(maze[hand_y][hand_x] == '.'){

            makeMove(??, ??, ??, ??);

        }
makeMove(x, y, hand_x, hand_y)
```



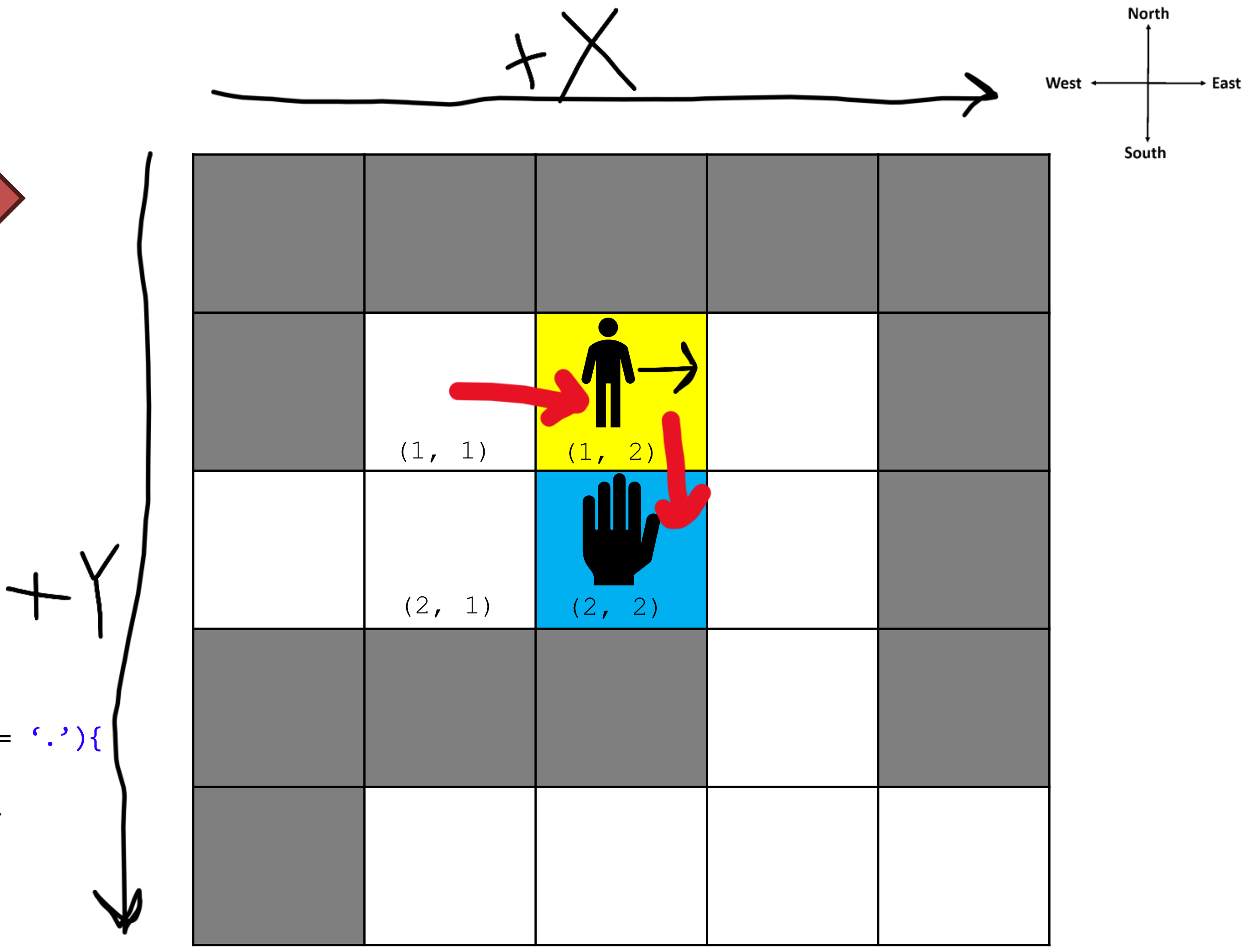
```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```

```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
        if(maze[hand_y][hand_x] == '.'){

            makeMove(x+1, y, hand_x, hand_y+1);

        }
makeMove(x, y, hand_x, hand_y)
```



```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```

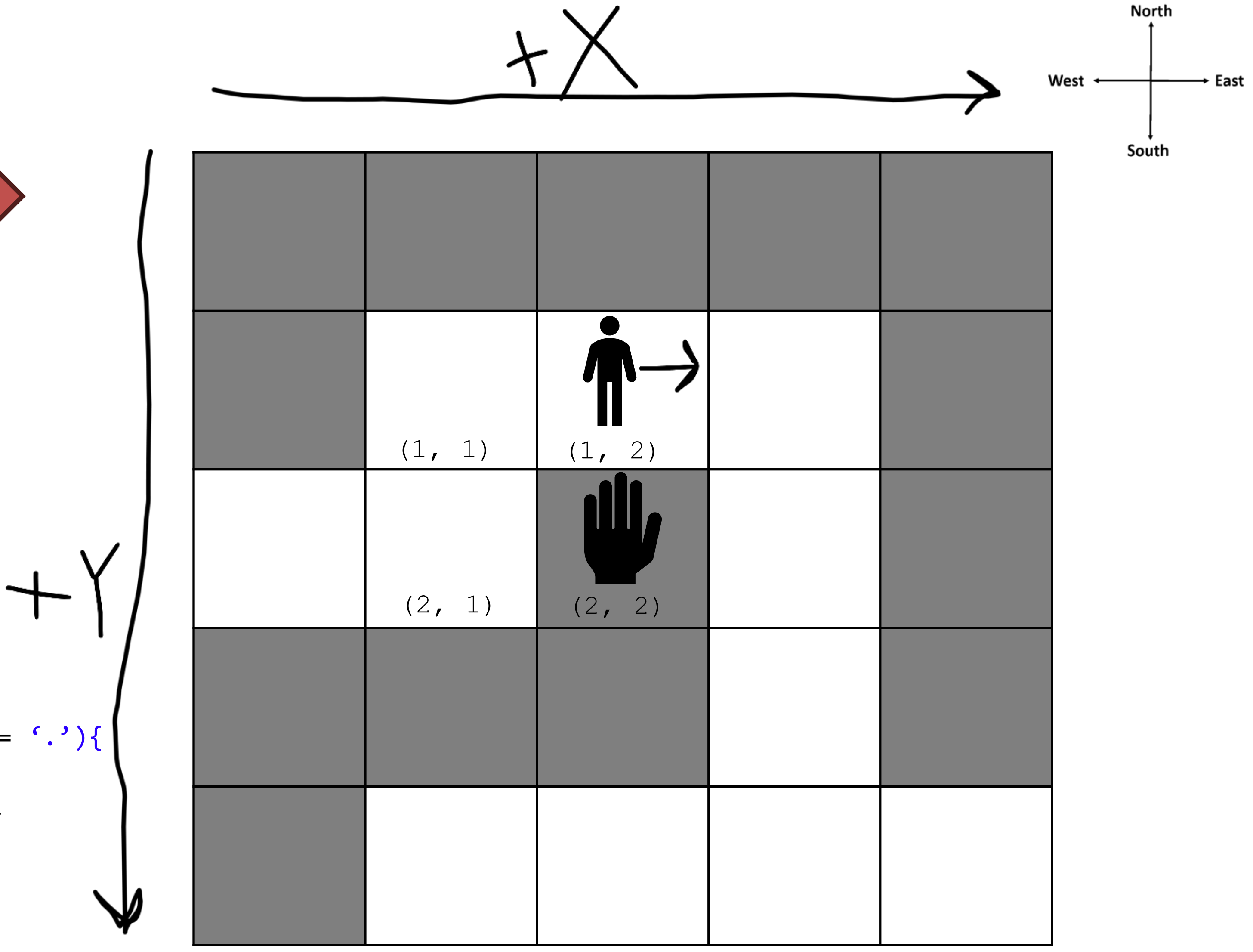
```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
        if(maze[hand_y][hand_x] == '.'){

            makeMove(x+1, y, hand_x, hand_y+1);

        }
makeMove(x, y, hand_x, hand_y)
```



maze[y][x]

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [ . , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```

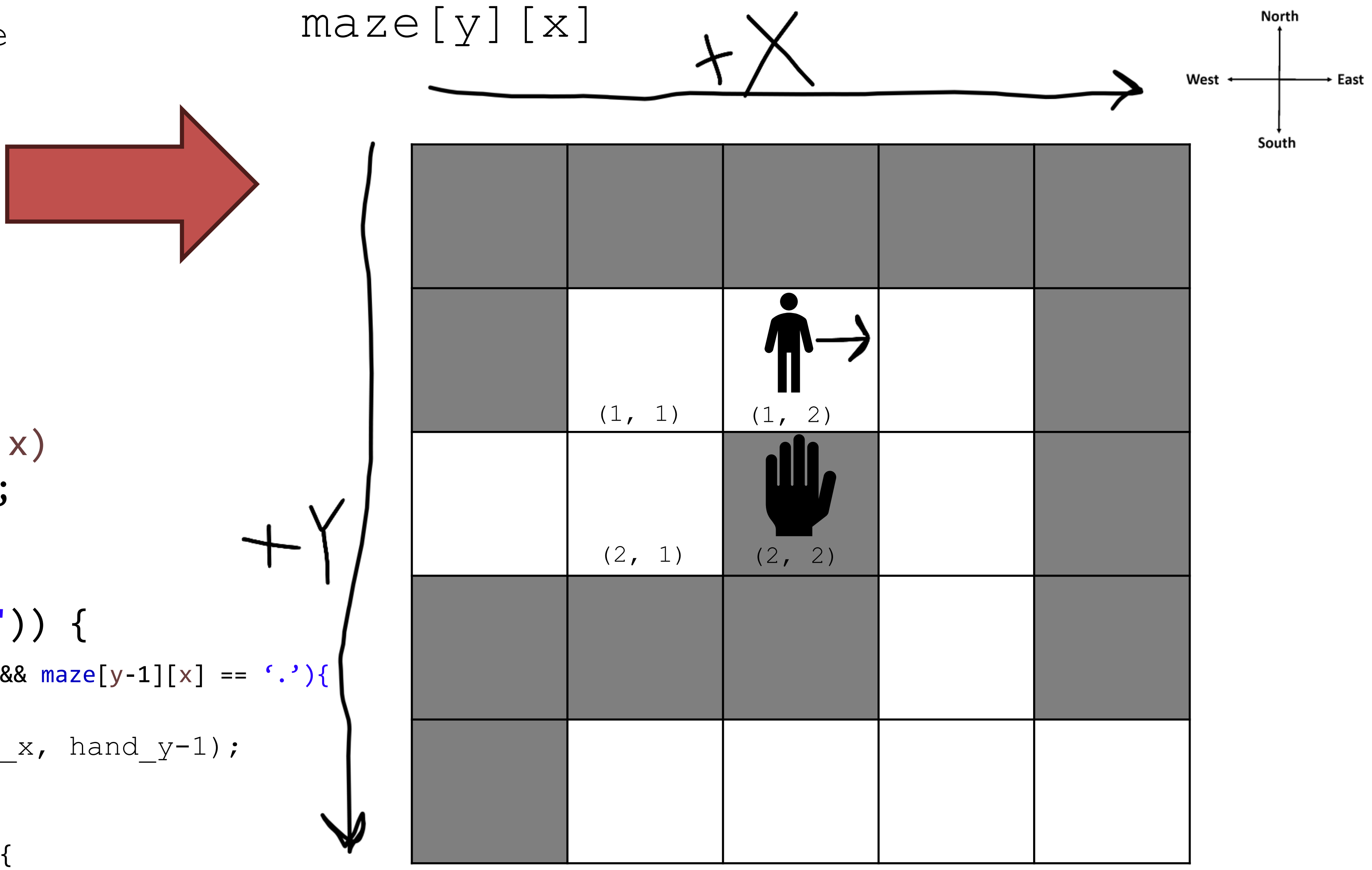
```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
        if(maze[hand_y][hand_x] == '.'){

            makeMove(x+1, y, hand_x, hand_y+1);

        }
makeMove(x, y, hand_x, hand_y)
```



maze[y][x]

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [ . , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```

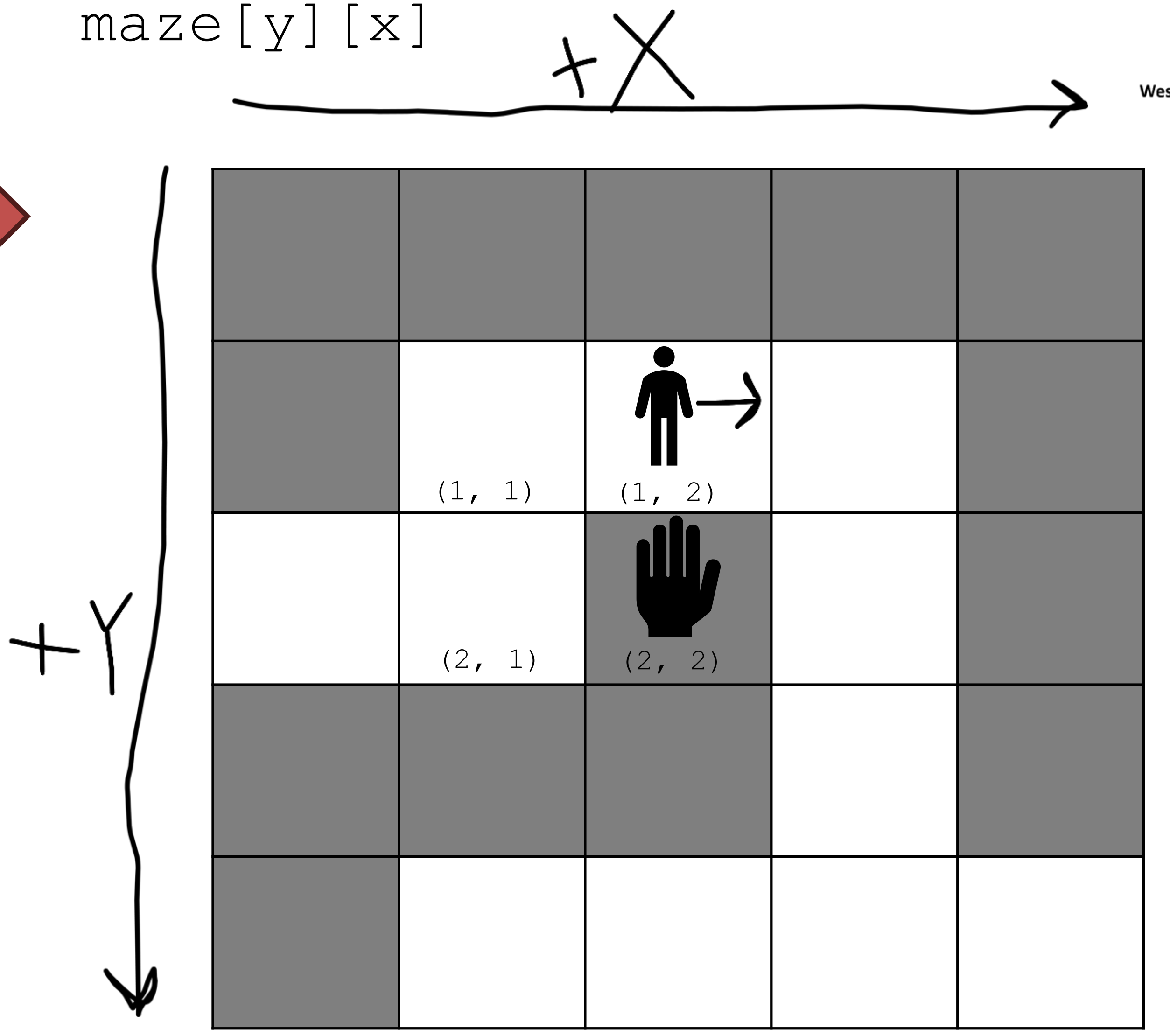
```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
        if(maze[hand_y][hand_x] == '.'){

            makeMove(x+1, y, hand_x, hand_y+1);

        }
makeMove(x, y, hand_x, hand_y)
```

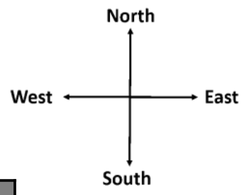


1. Turn right
2. Go forward
3. Turn left

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [ . , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , . ],
]
```

maze[y][x]





```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
      if(maze[hand_y][hand_x] == ‘.’){
         makeMove(x+1, y, hand_x, hand_y+1);

      }

      if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == ‘.’){

         makeMove(x, y-1, hand_x, hand_y-1);

      }
```

Right

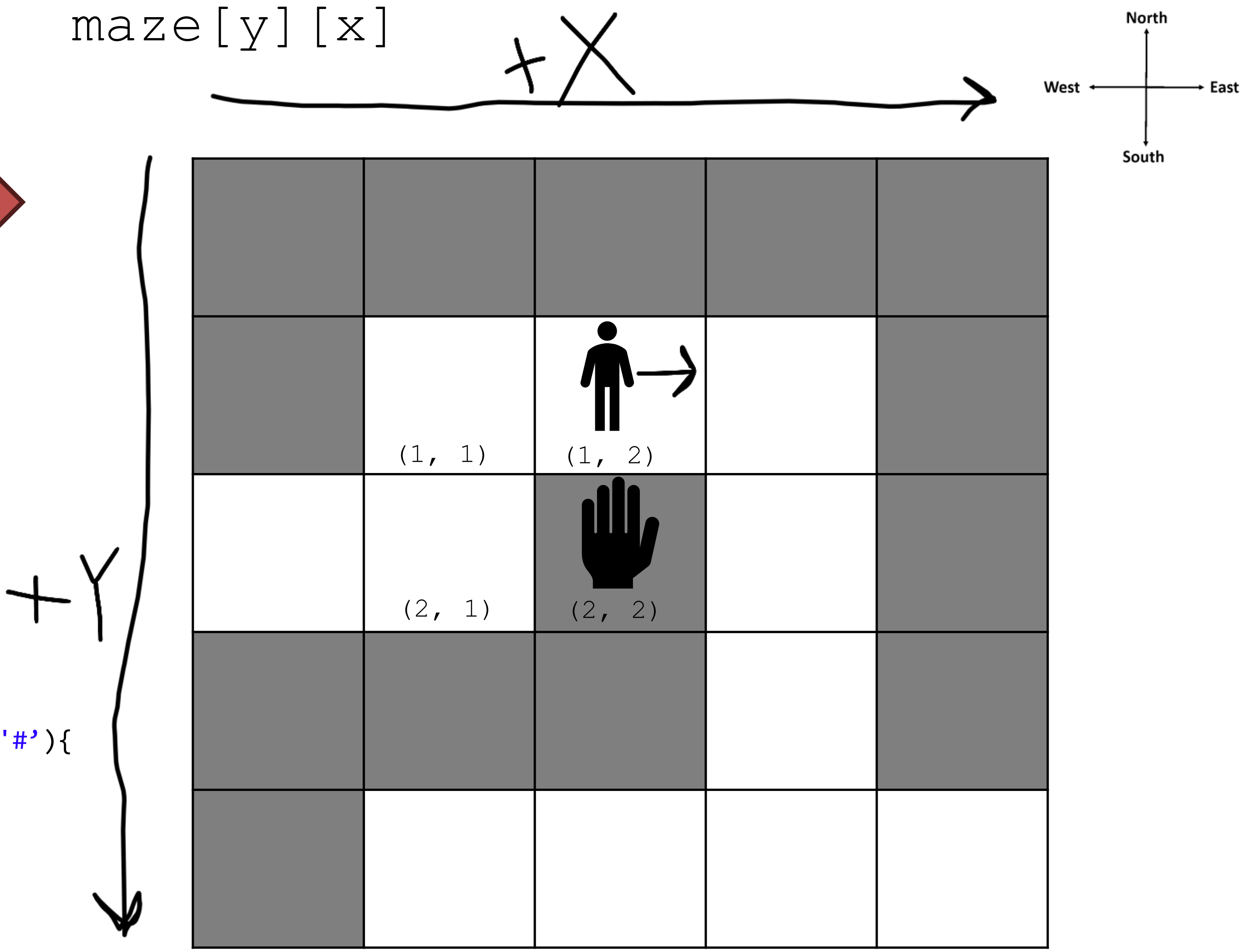
Forward

Left

// Turn left

1. Turn right
2. Go forward
3. Turn left

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , .],
]
```



```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
        if(maze[hand_y][hand_x] == '.' && maze[y-1][x]=='#'){
            makeMove(x+1, y, hand_x, hand_y+1);

        }

        if(maze[hand_y][hand_x] == '#' && maze[y-1][x] == '.'){

            makeMove(x, y-1, hand_x, hand_y-1);

        }
        // Turn left
```

Right

Forward

Left

You will have need if statements for North, East, South, and West

Lots of if statements ☺

```
char[][] maze

[[#, #, #, # ,#],
 [#, . , . , . , #],
 [. , . ,# , . , #],
 [#, #, #, . , #],
 [#, . , . , . , .],
]
```

```
if(y == hand_y && hand_x > x)
        direction = "North";
}
…
if(direction.equals("North")) {
    if(maze[hand_y][[        aze[y-1][x]=='#’){
       make                          +1);

    }

    if(ma                            == ‘.’){
       make                          -1);

    }
// Turn left
```

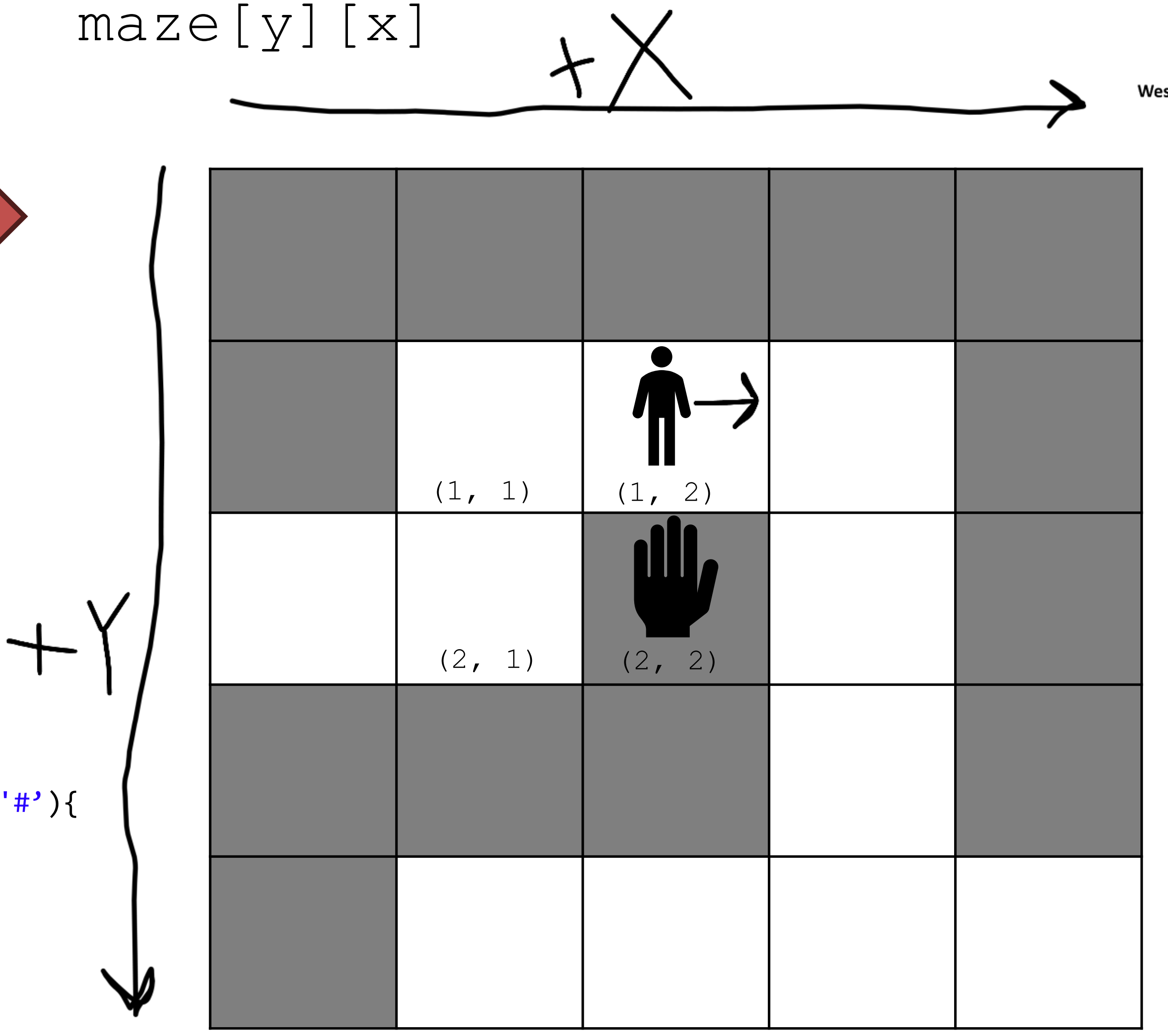
Right

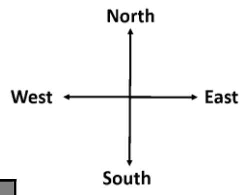
Forward

Left

This code is technically not complete, you will need to add some more code here (backtracking)

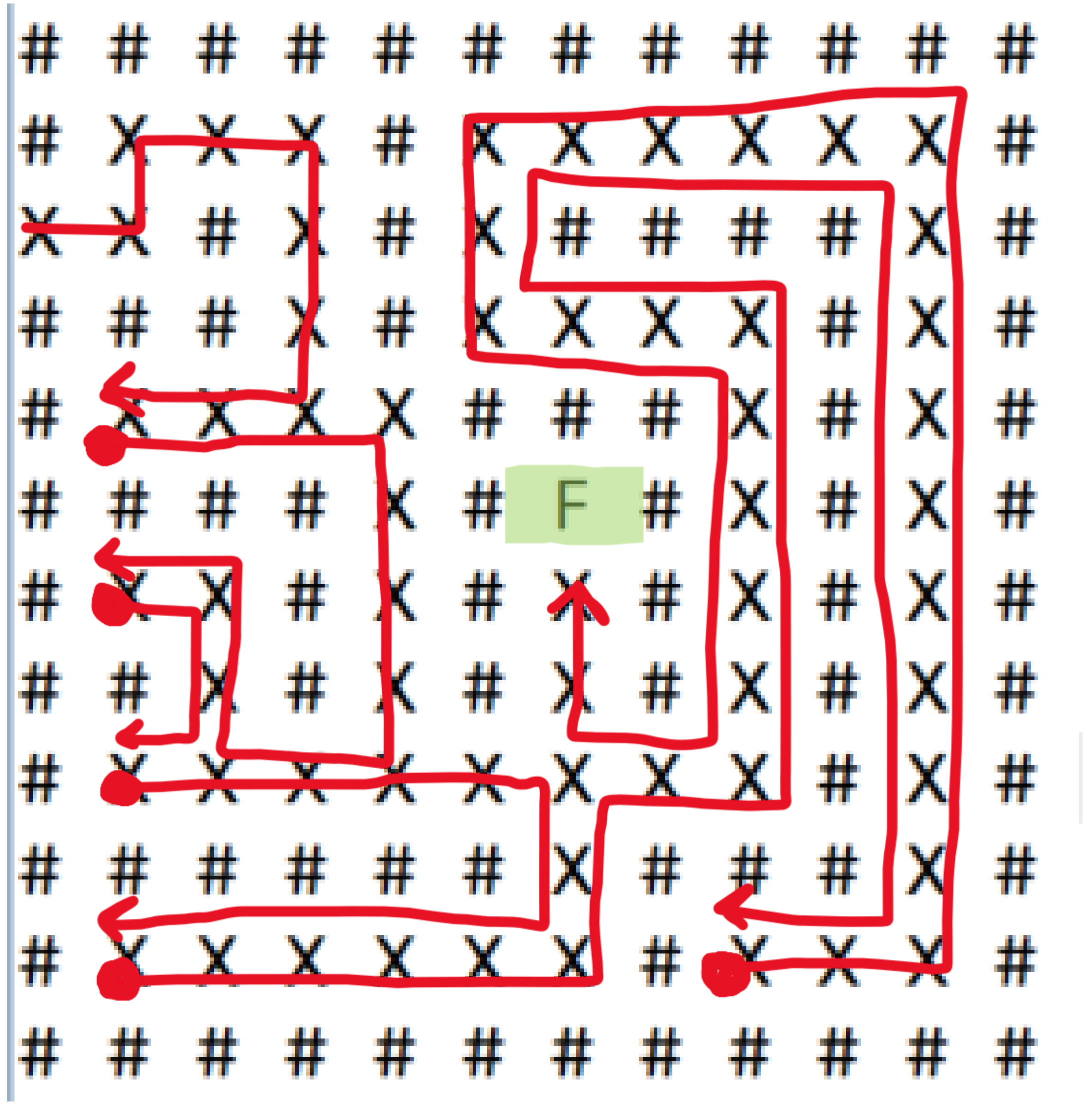
maze[y][x]





You will have need if statements for North, East, South, and West

Lots of if statements ☺



```
# # # # # # # # # # # #
# X X X # X X X X X X #
X X # X # X # # # # X #
# # # X # X X X X # X #
# X X X X # # # X # X #
# # # # X # F # X # X #
# X X # X # X # X # X #
# # X # X # X # X # X #
# X X X X X X X X # X #
# # # # # # X # # # X #
# X X X X X X # X X X #
# # # # # # # # # # # #
```

● = Backtracking path