You are in a maze of twisty passages...

Minos 2009

Peter Harrison

www.micromouseonline.com

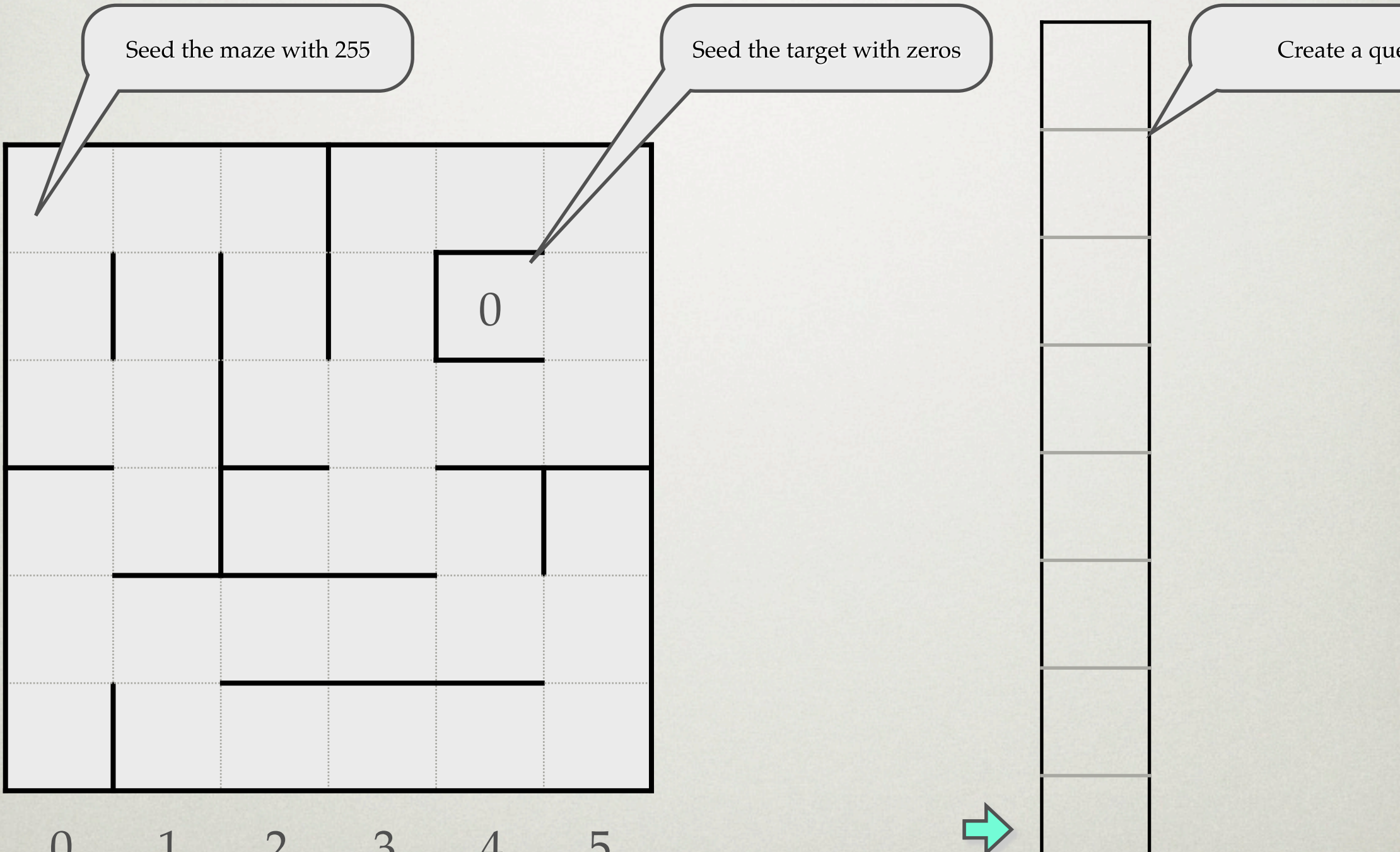
Fast is good

- Exploration Penalty
- Non-stop Exploring
- Multiple Strategies

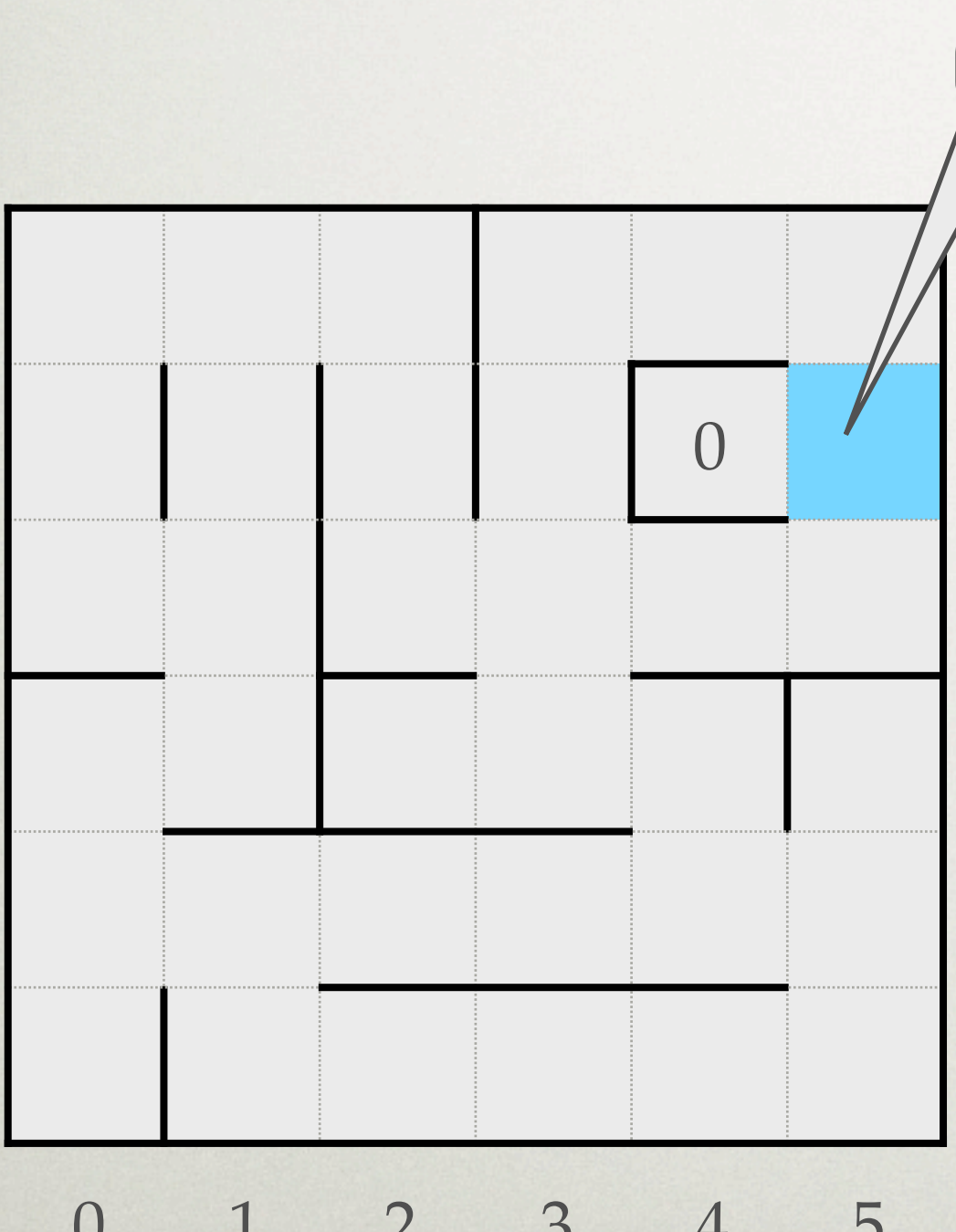
Fast is good

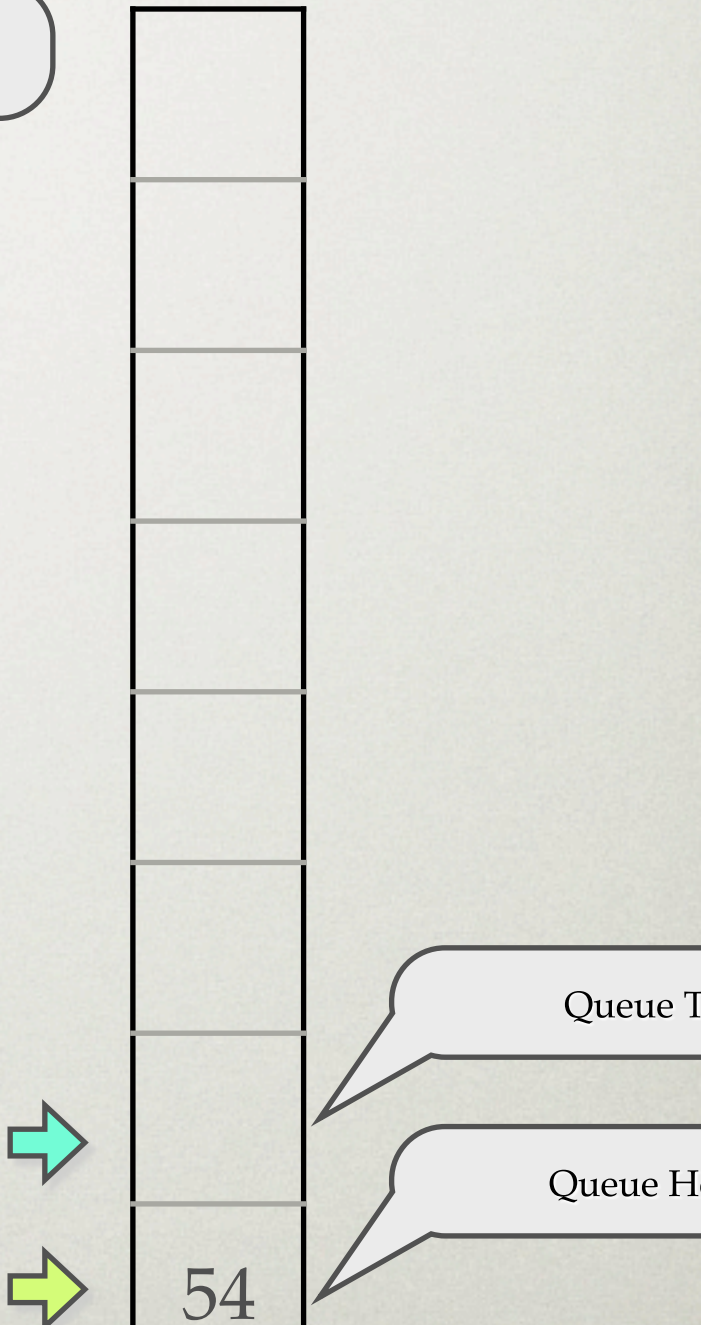
- Explore: $v = 0.5ms^{-1}$, $a = 1.5ms^{-2}$
- Braking distance $s_b = v^2/2a = 83mm$
- Decide on cell boundary
- Thinking distance = 7mm
- Thinking time = s/v = 14ms

AT THE START

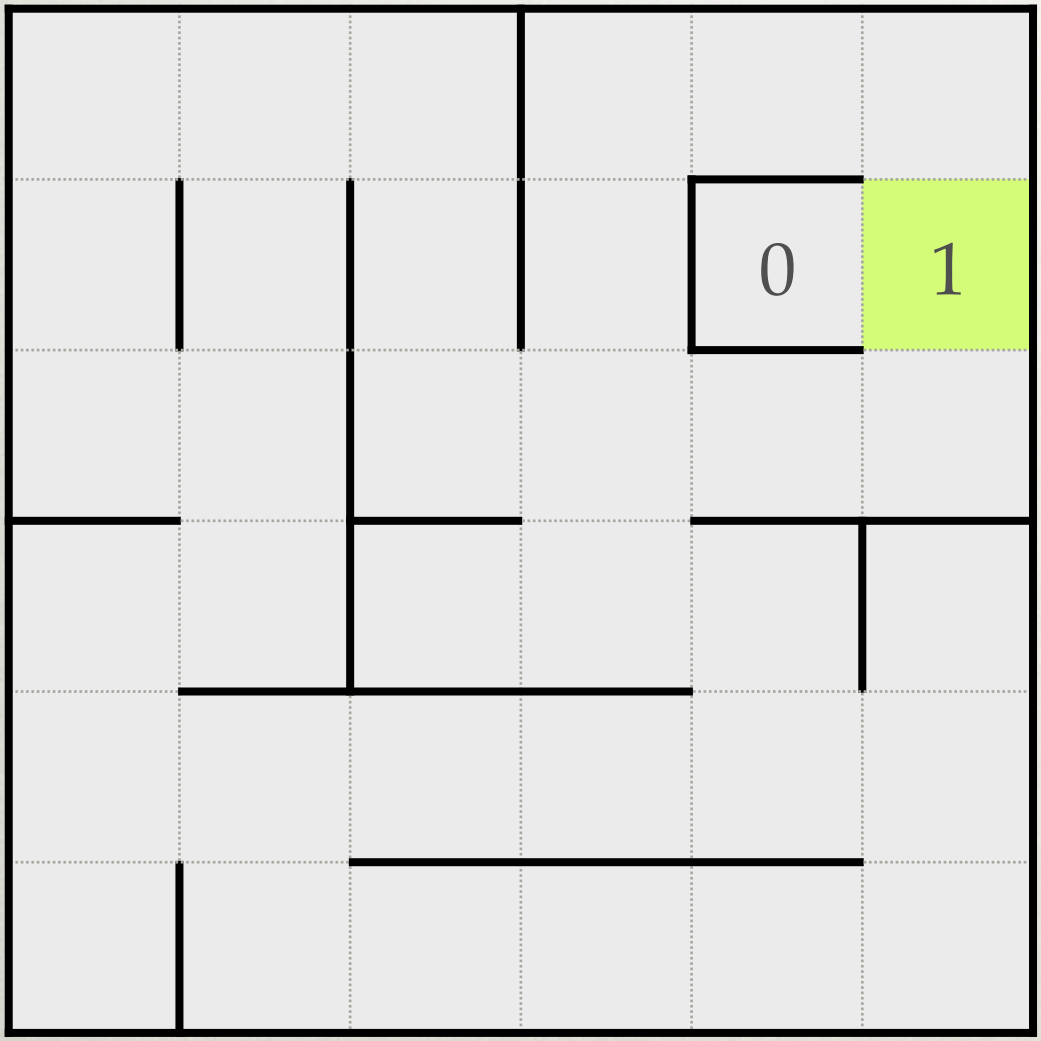

Seed the maze with 255
Seed the target with zeros
Create a que
0
0
1
2
3
4
5

READY TO GO

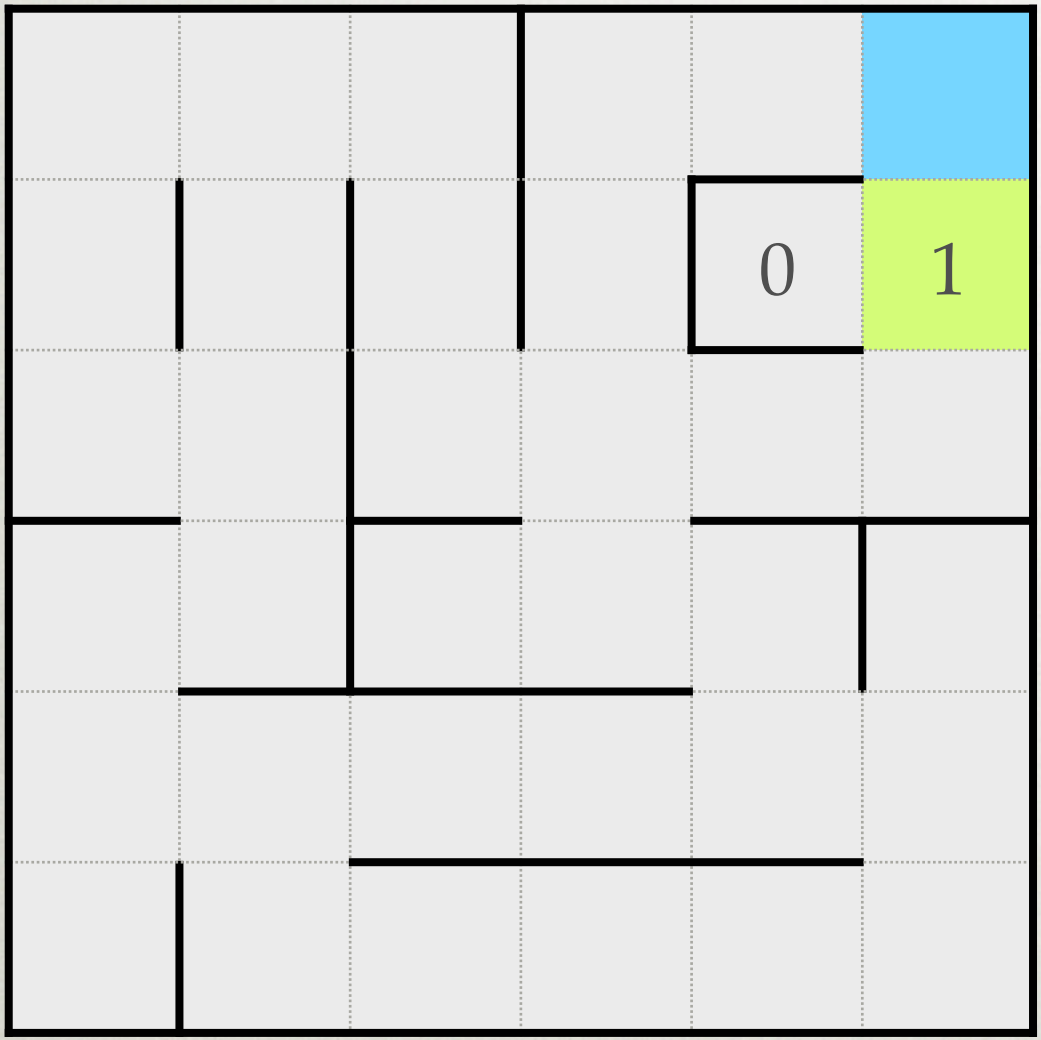




CALCULATE DISTANCE

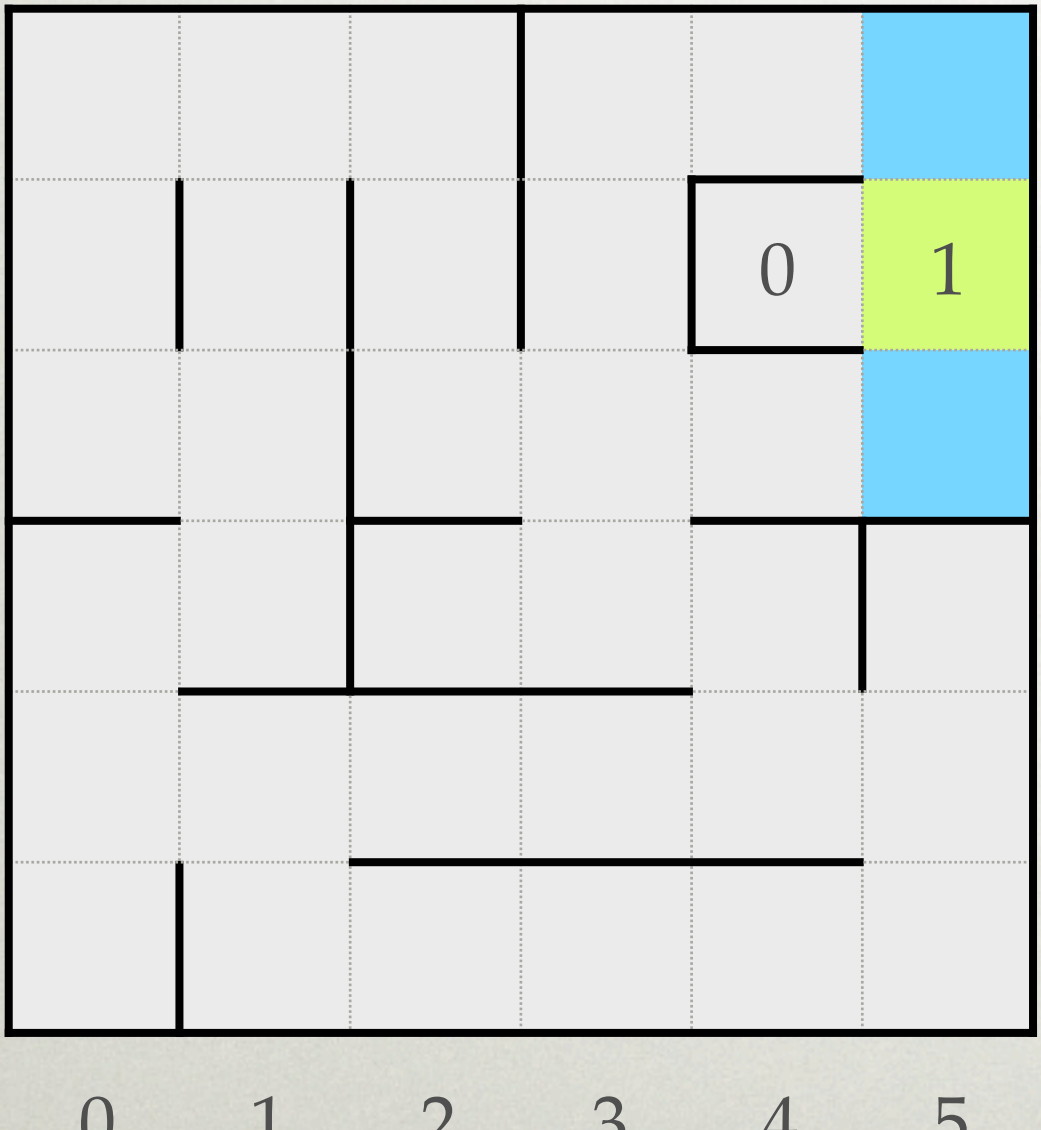


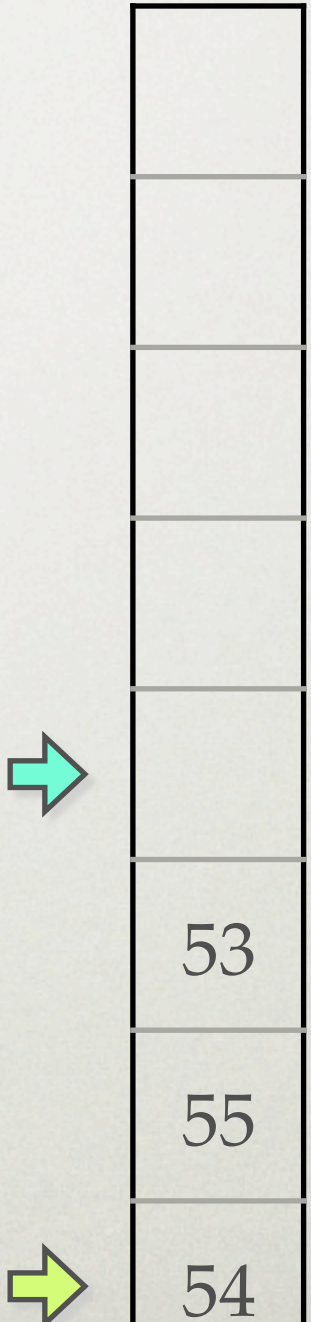
Add neighbours



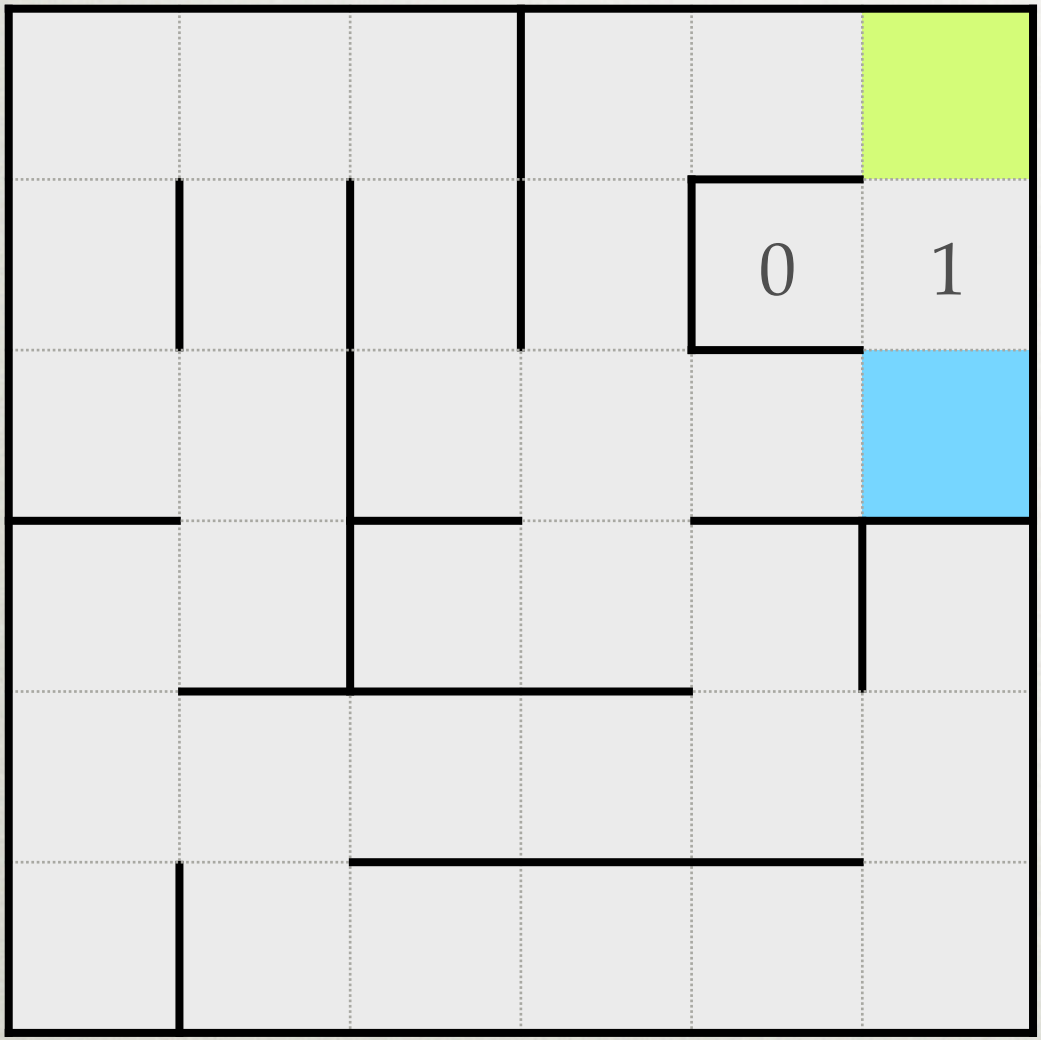
clockwise from

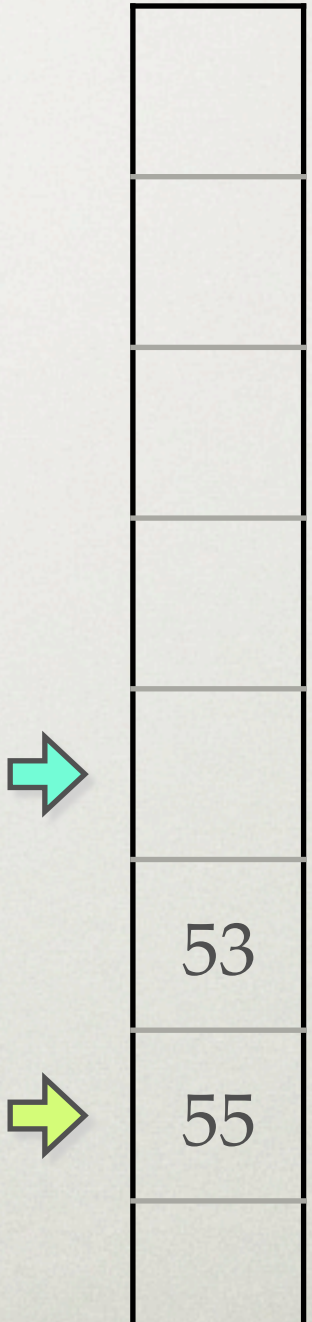
Add neighbours



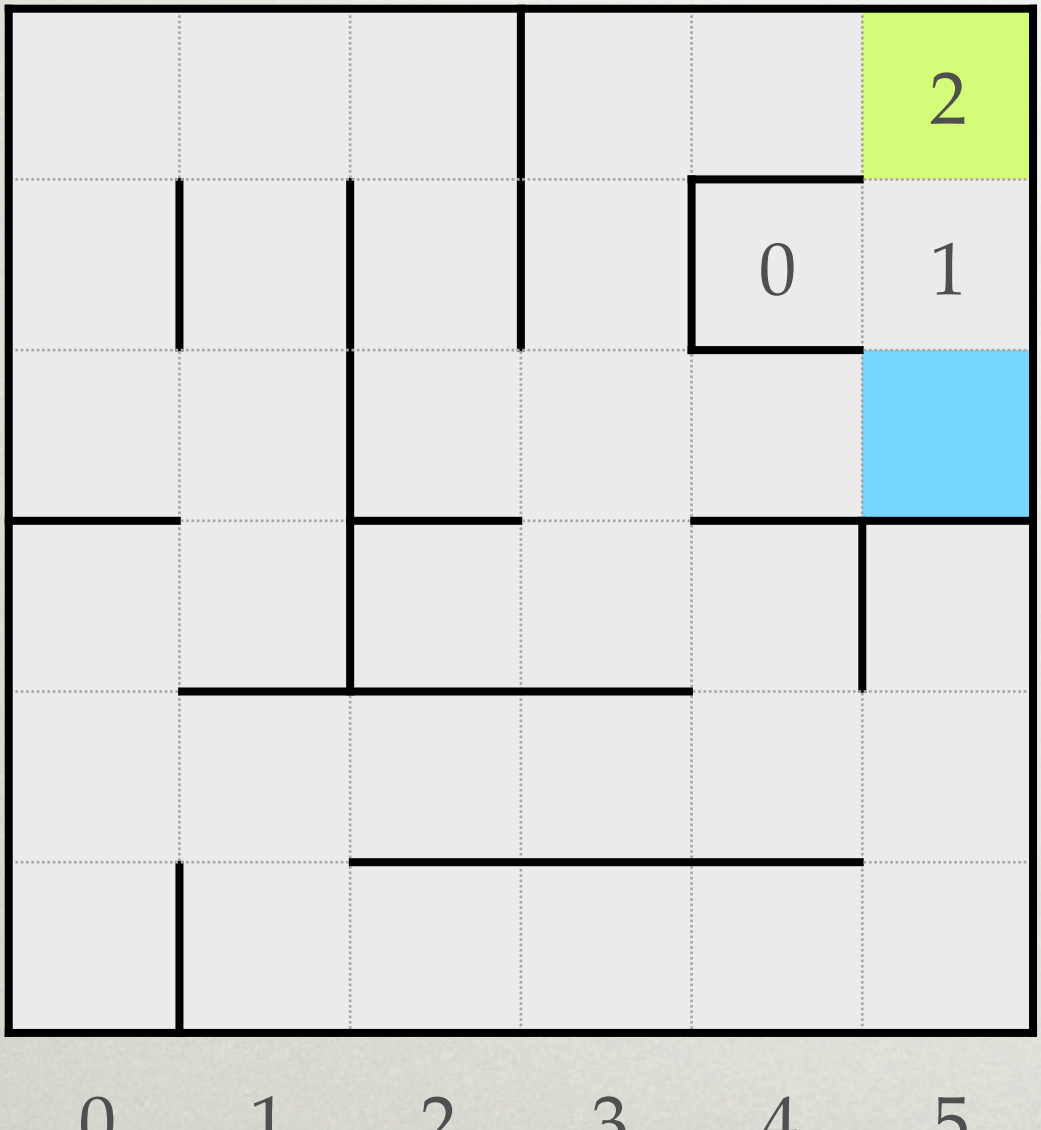


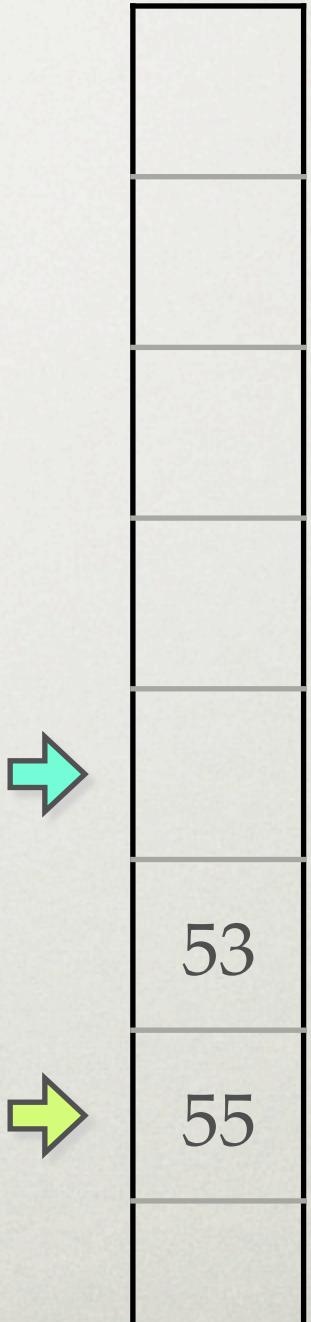
FINISHED PROCESSING



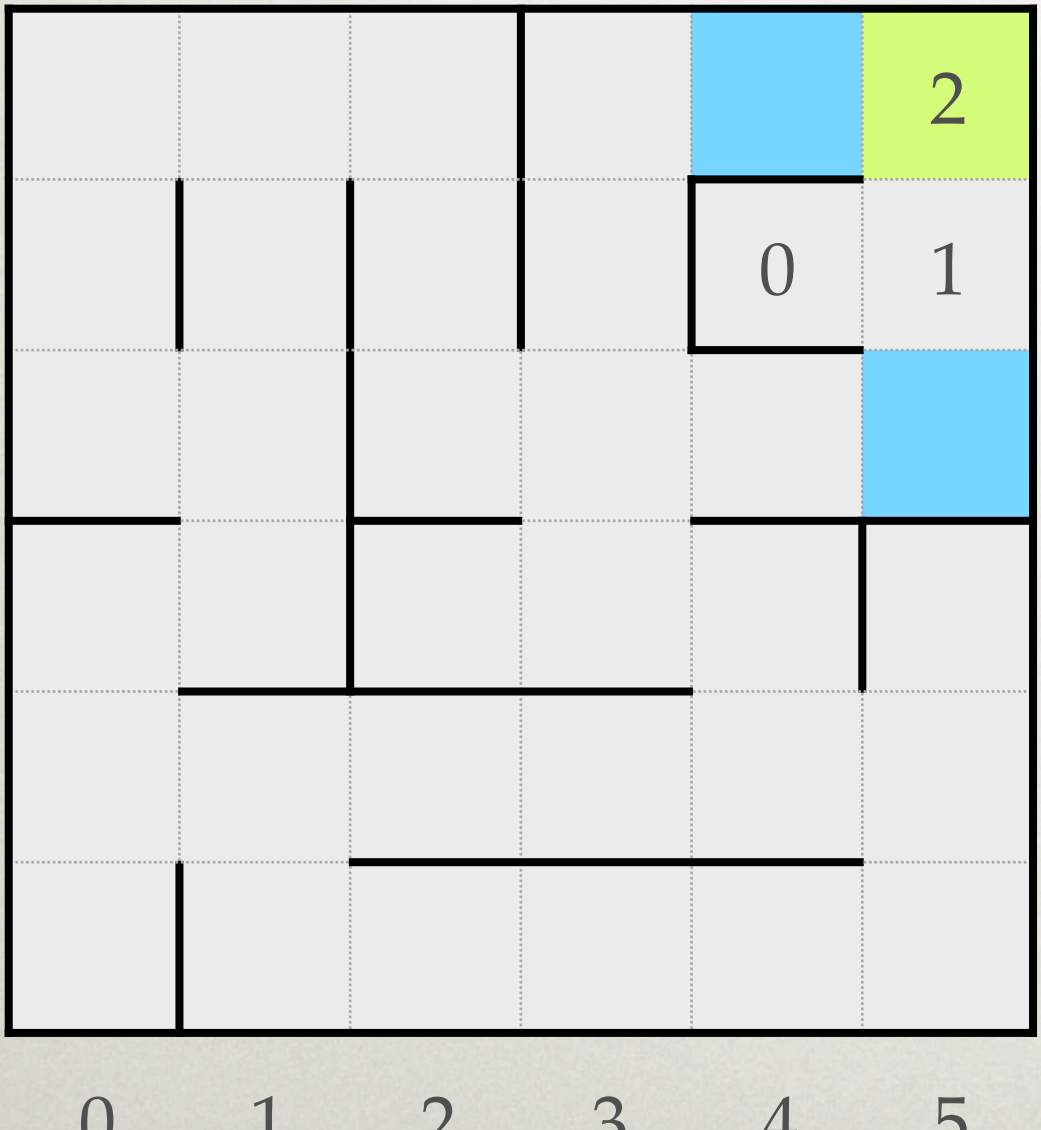


CALCULATE DISTANCE

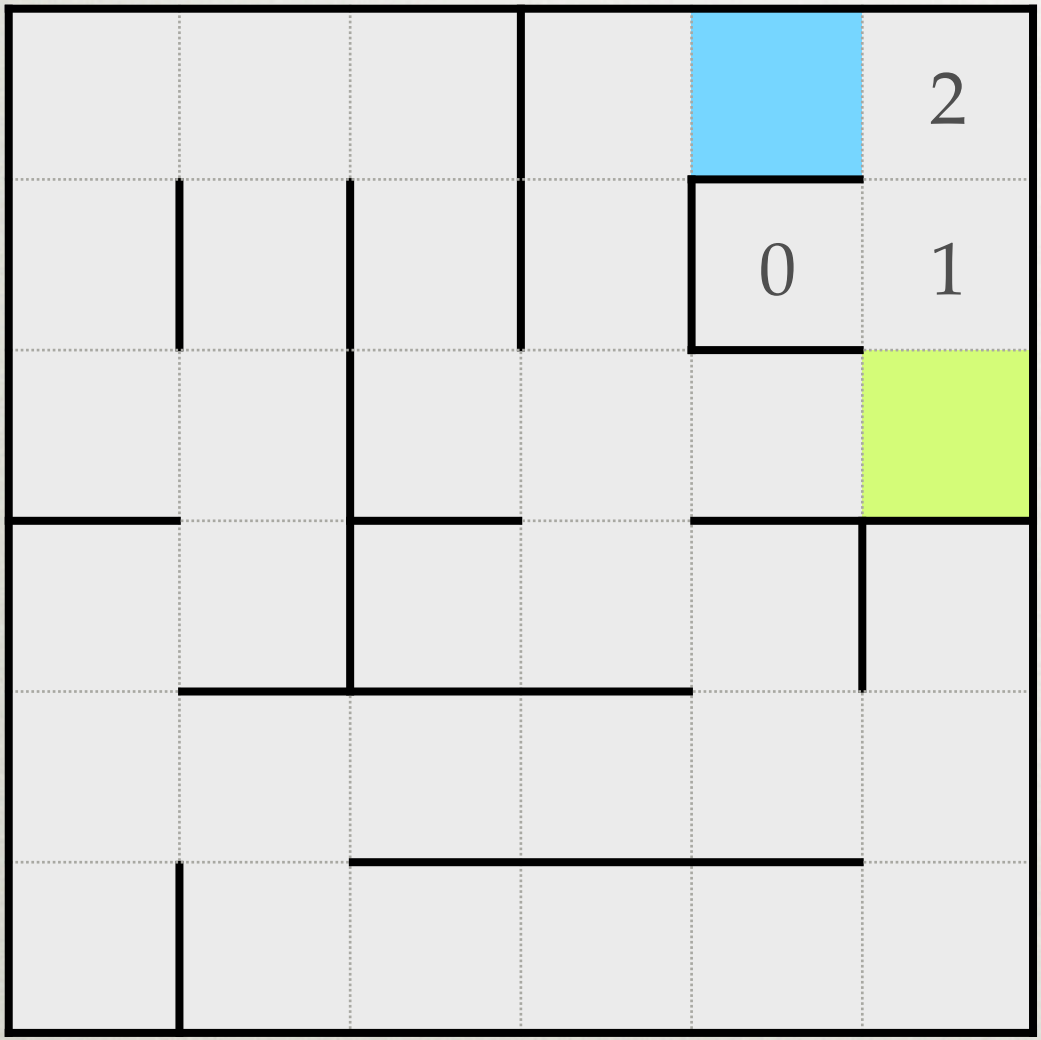




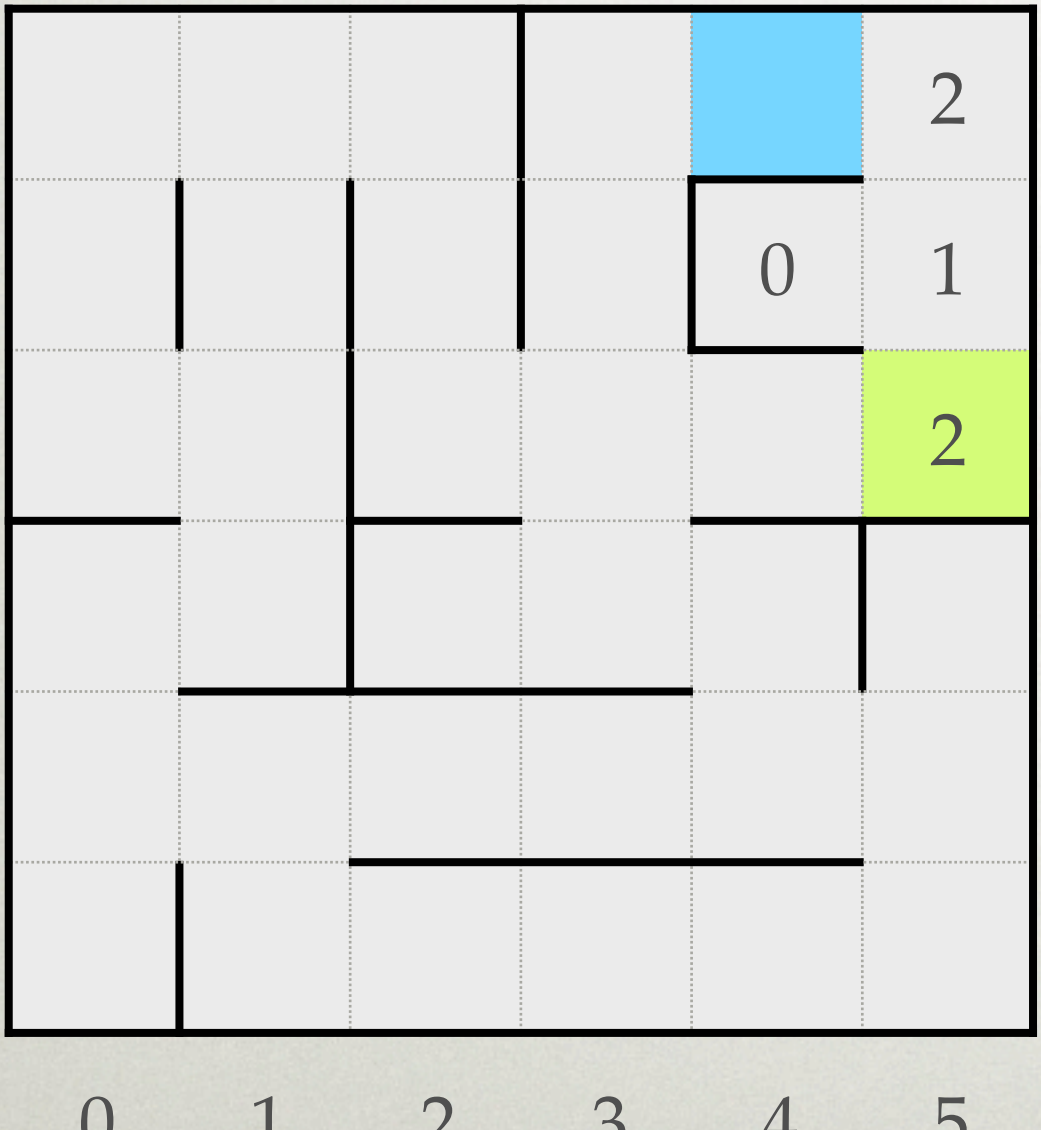
Add neighbours

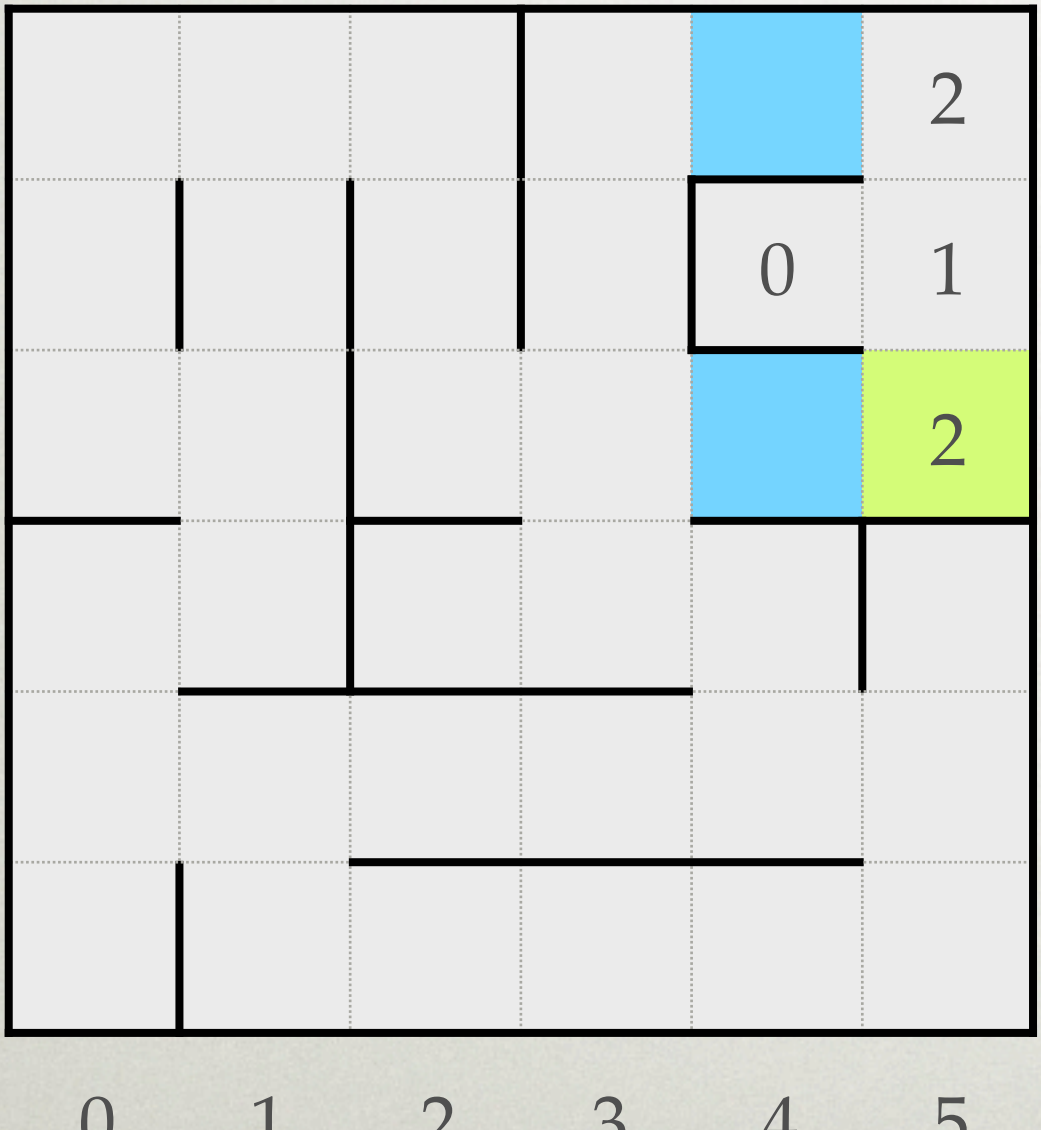


MOVE HEAD

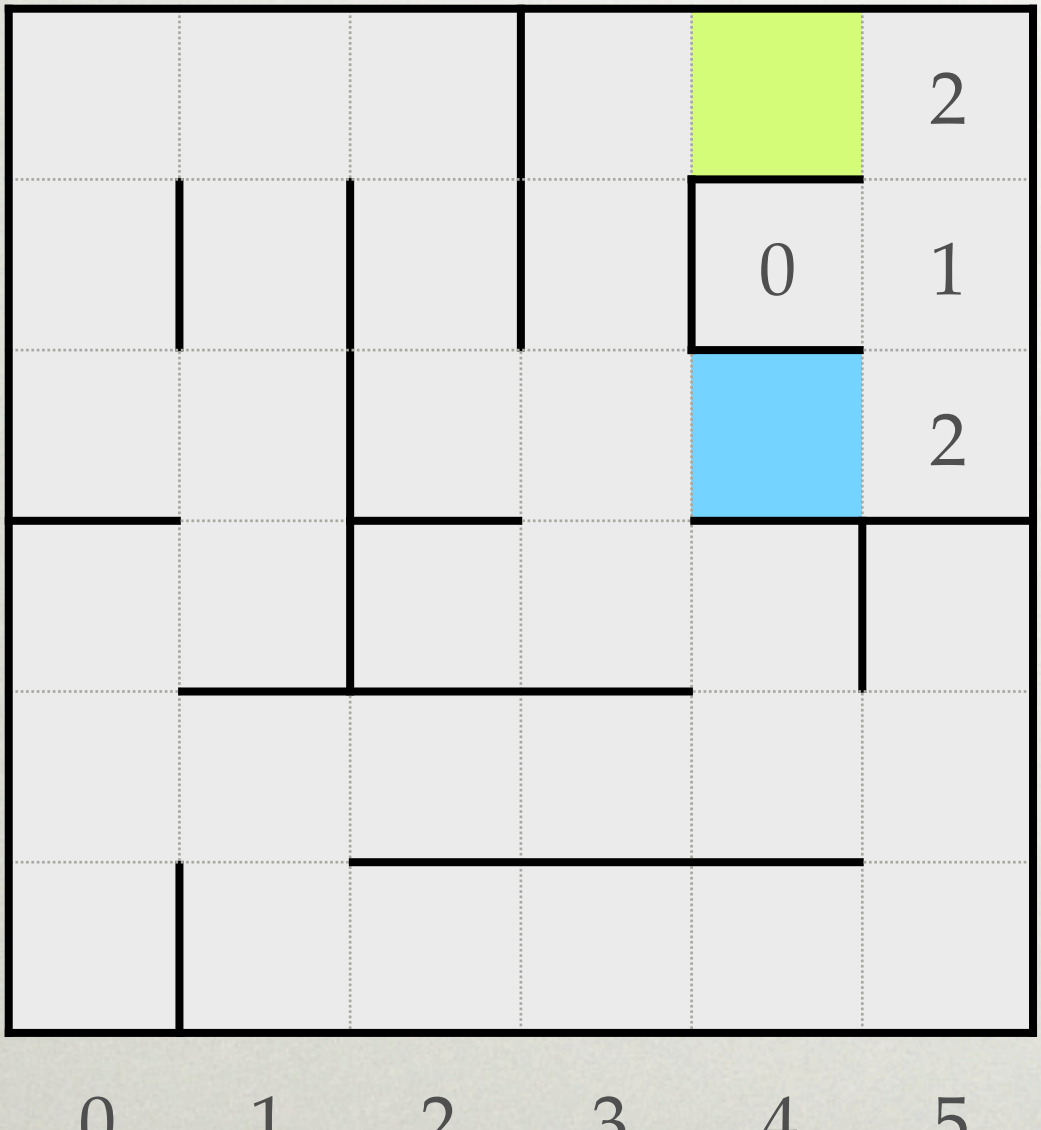


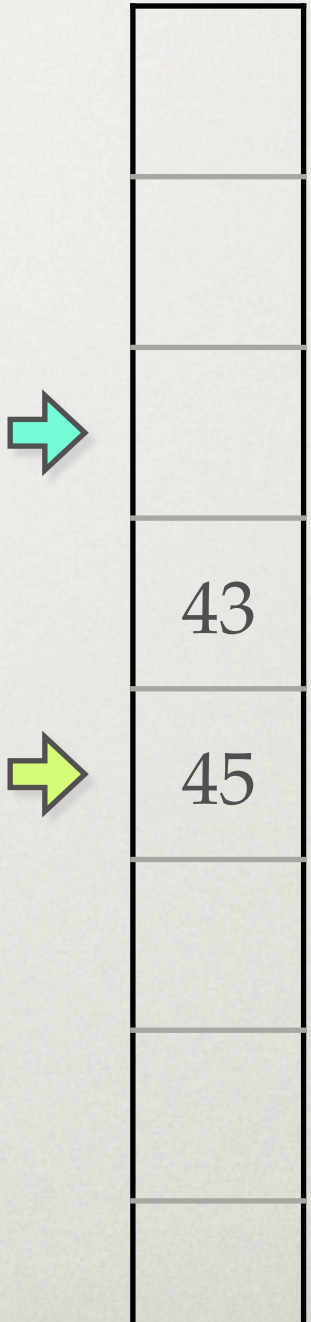
CALCULATE DISTANCE

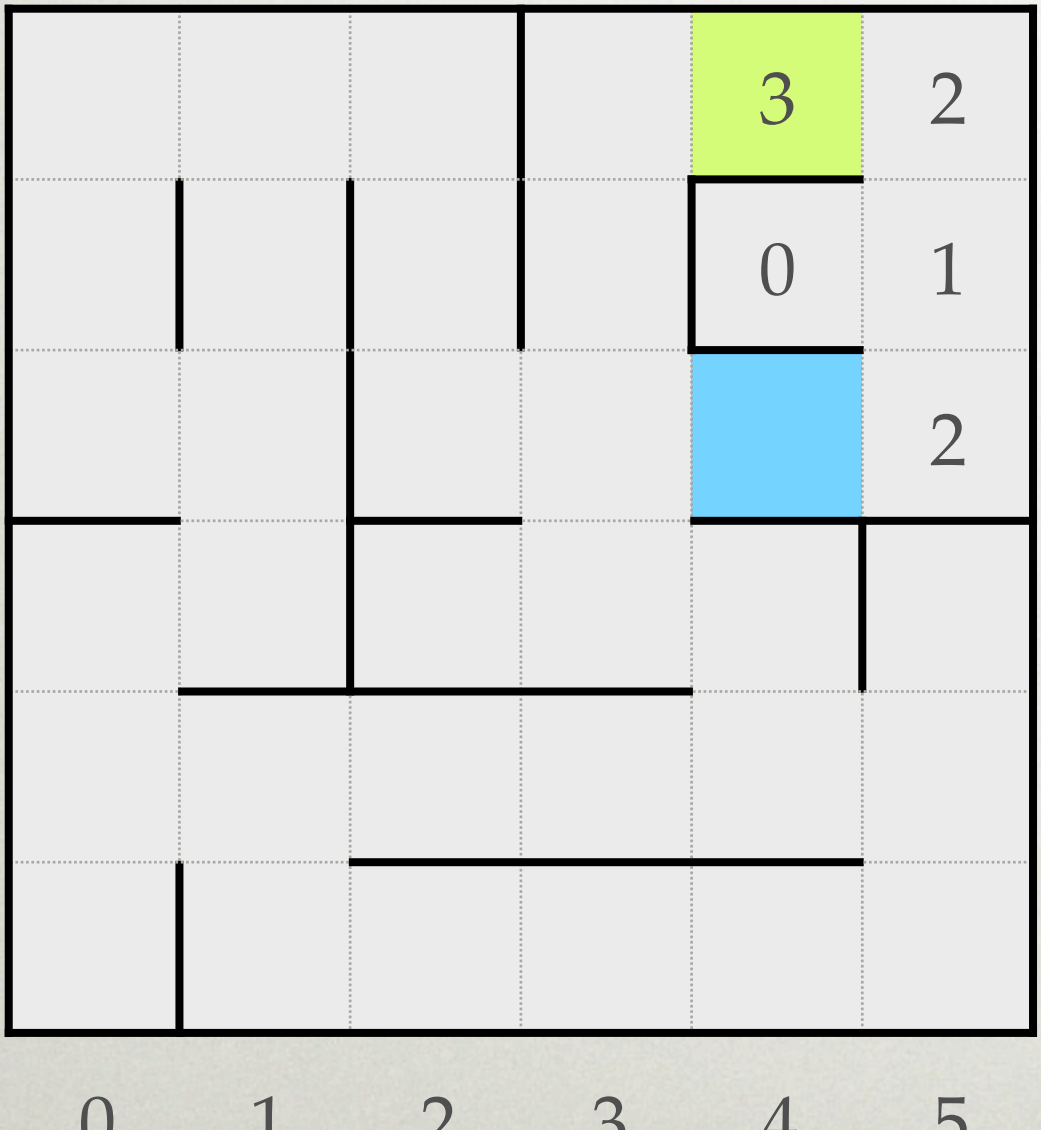


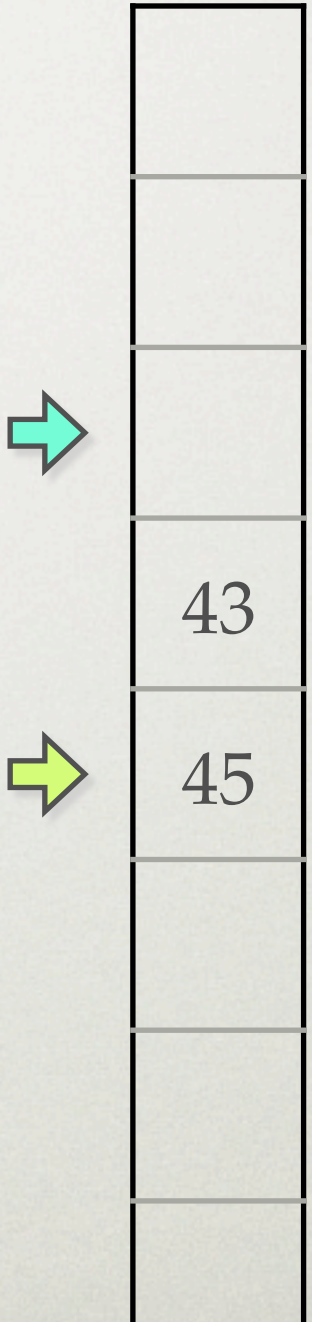

2
0
1
2
0
1
2
3
4
5

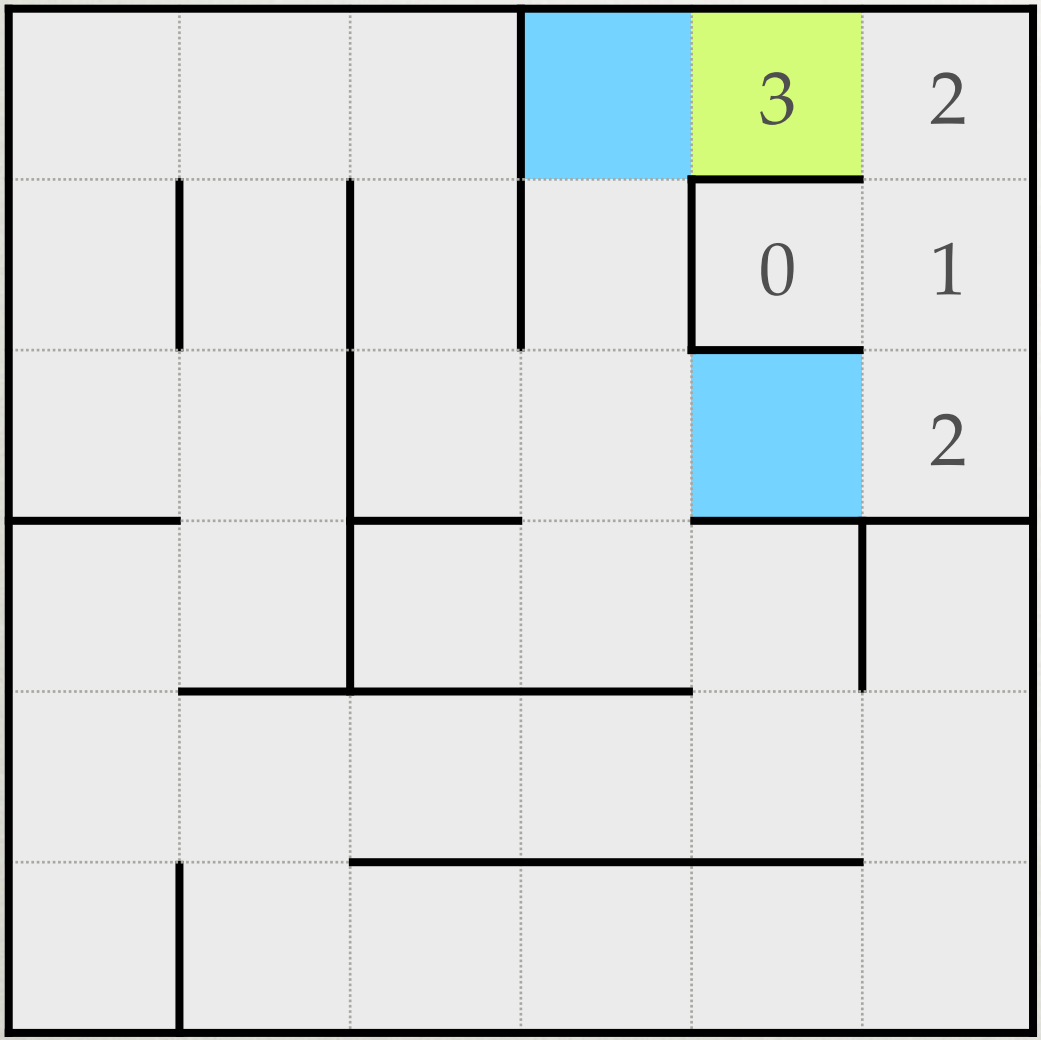

43
45
53

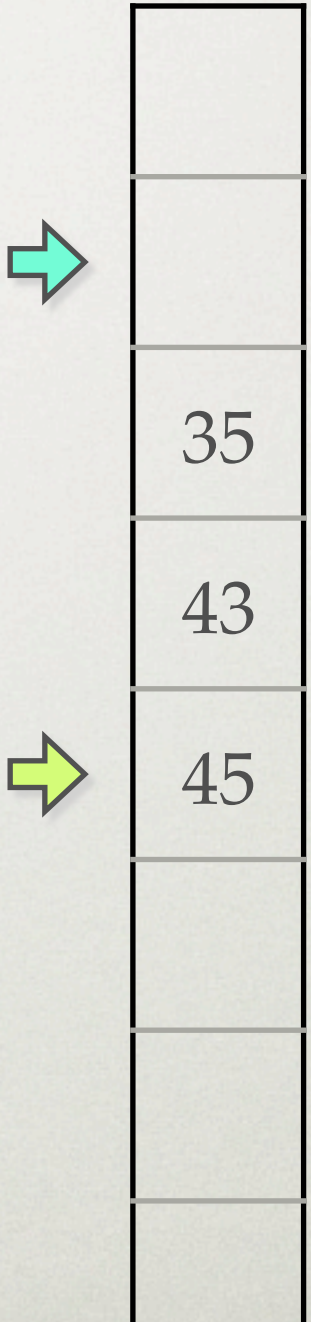

2
0
1
2
0
1
2
3
4
5

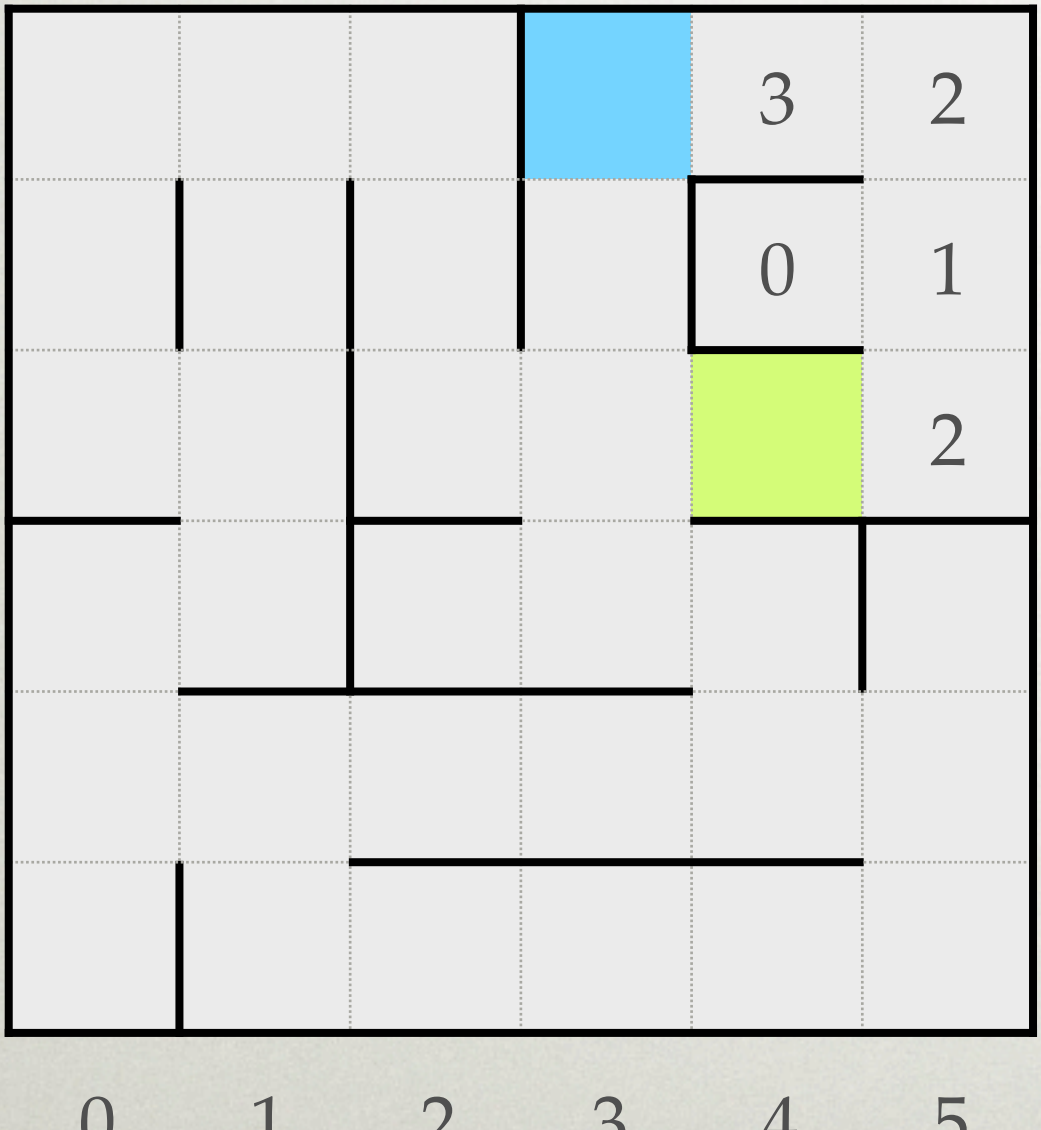

43
45


3
2
0
1
2
0
1
2
3
4
5


43
45

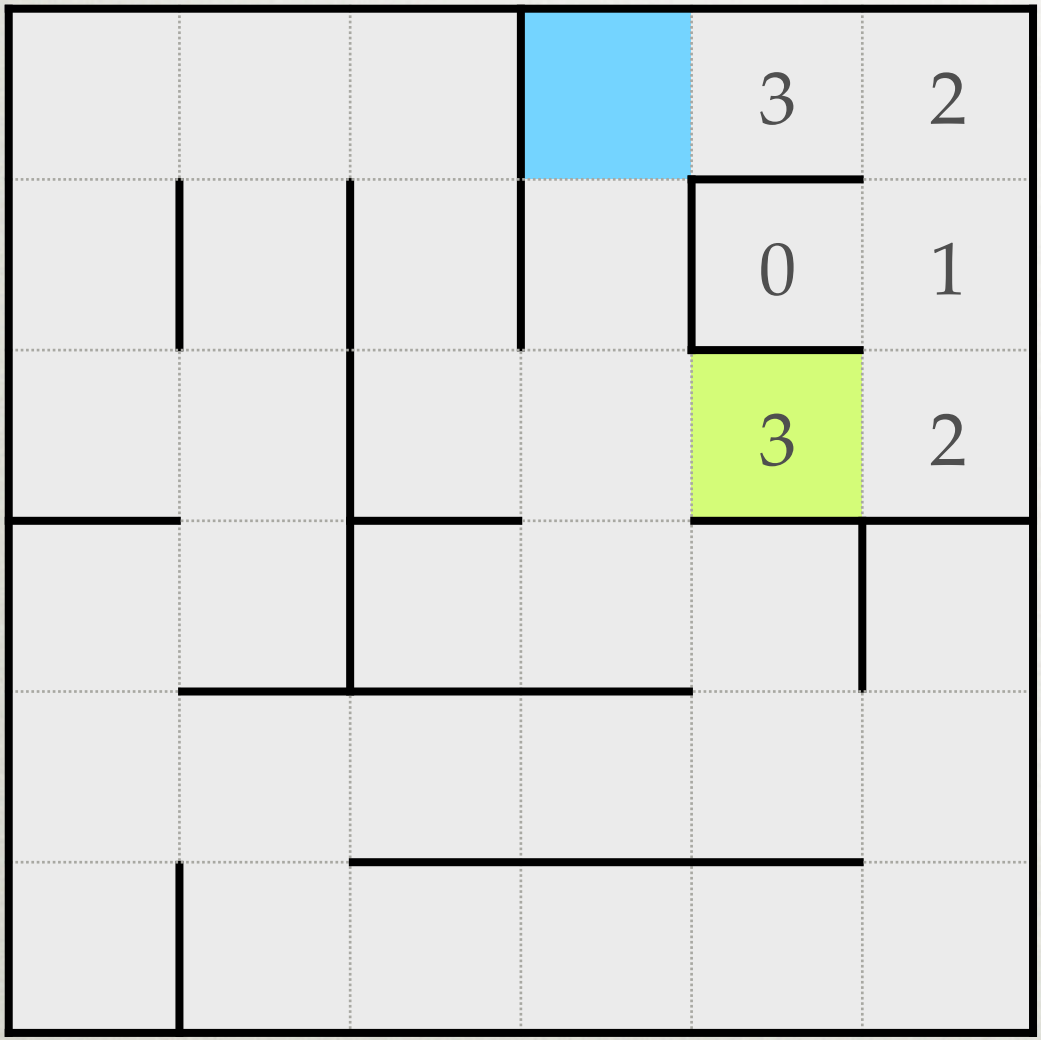

3
2
0
1
2
0
1
2
3
4
5


35
43
45

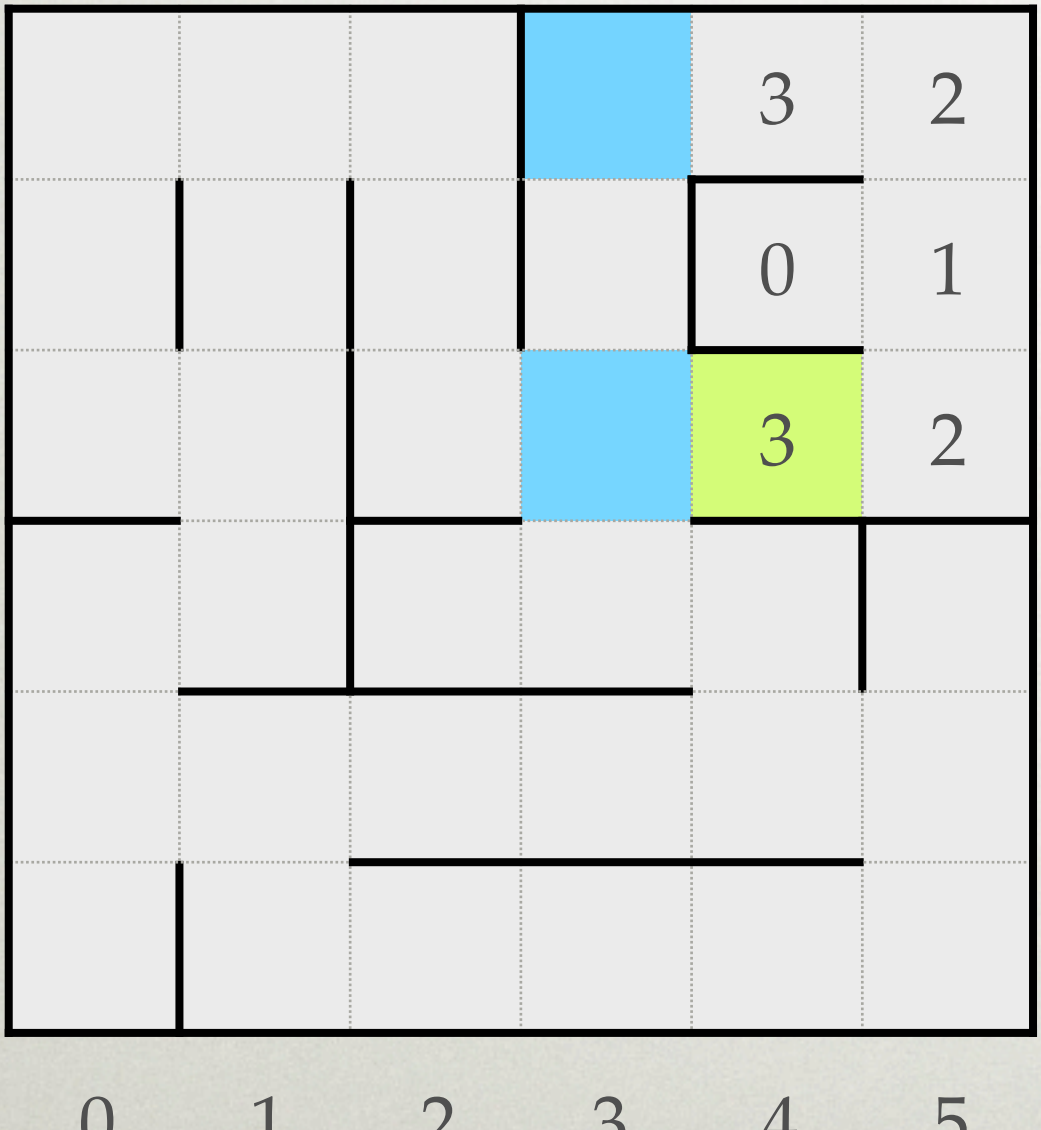

3
2
0
1
2
0
1
2
3
4
5


35
43

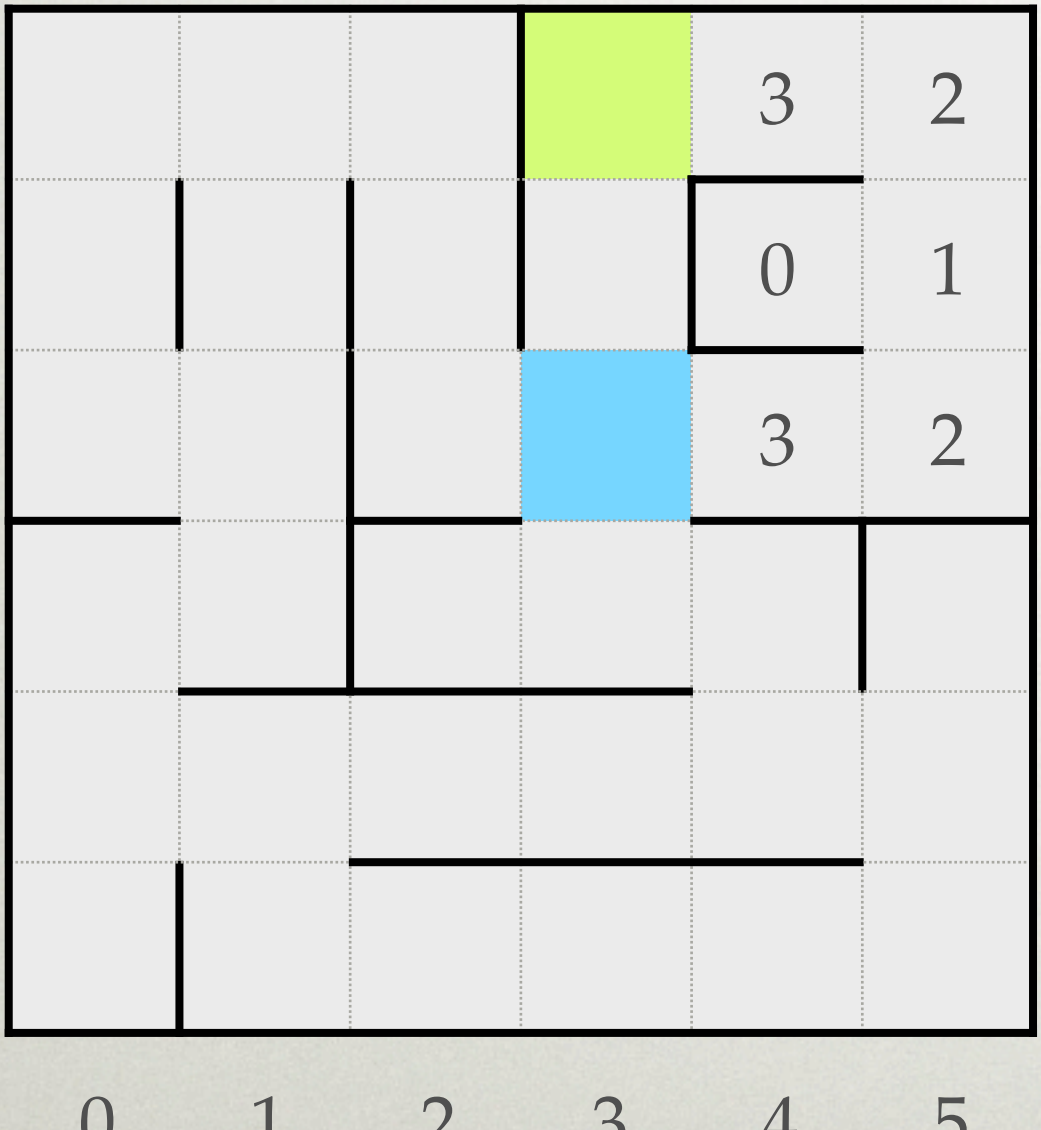
DISTANCE



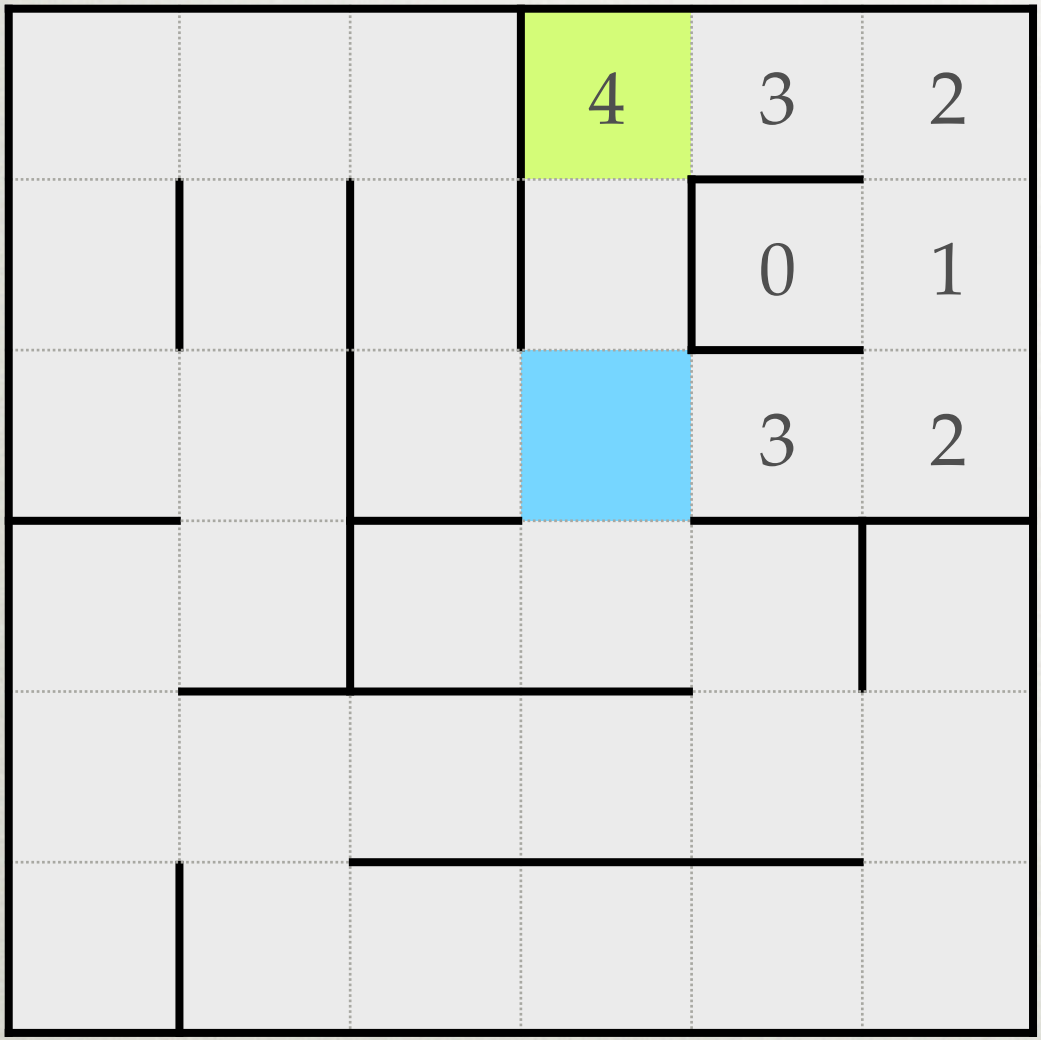
NEIGHBOURS



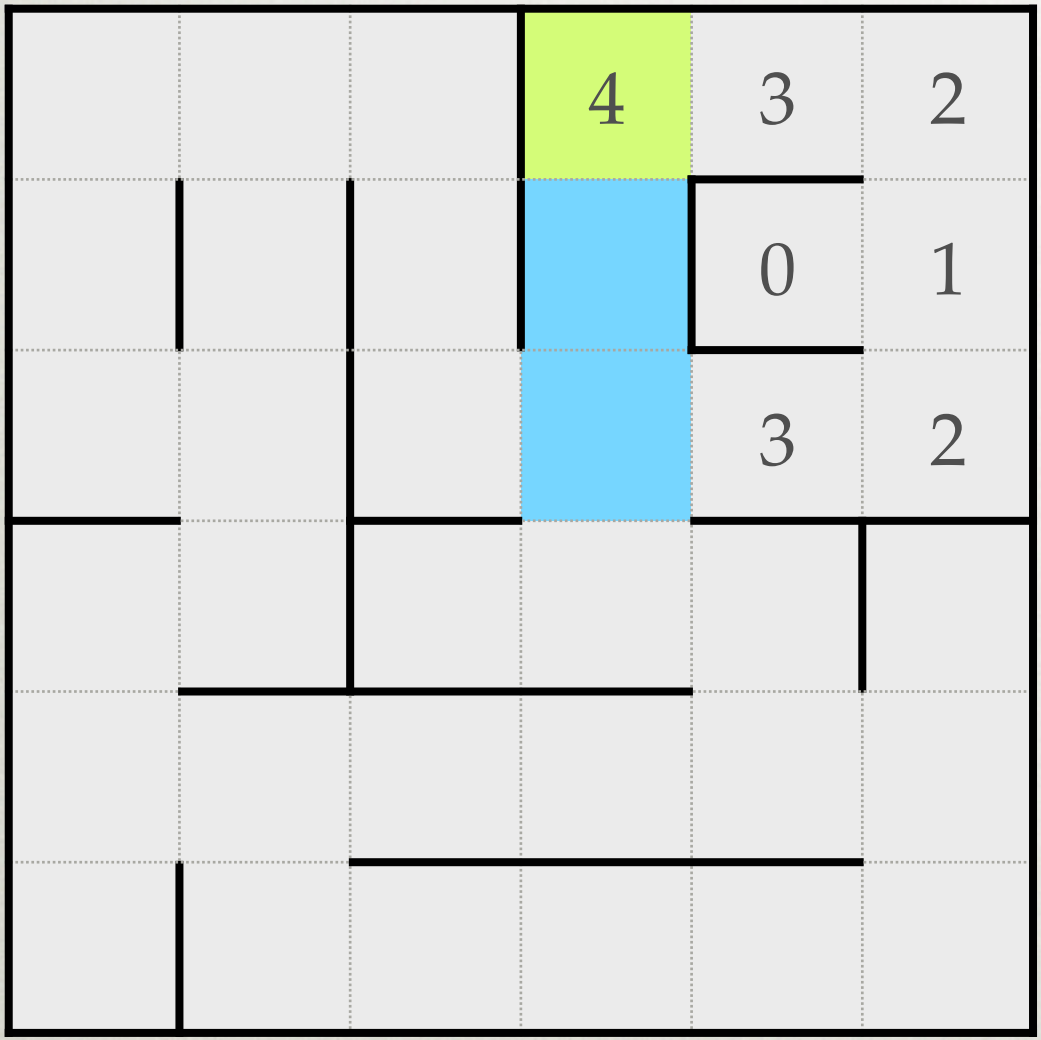
HEAD



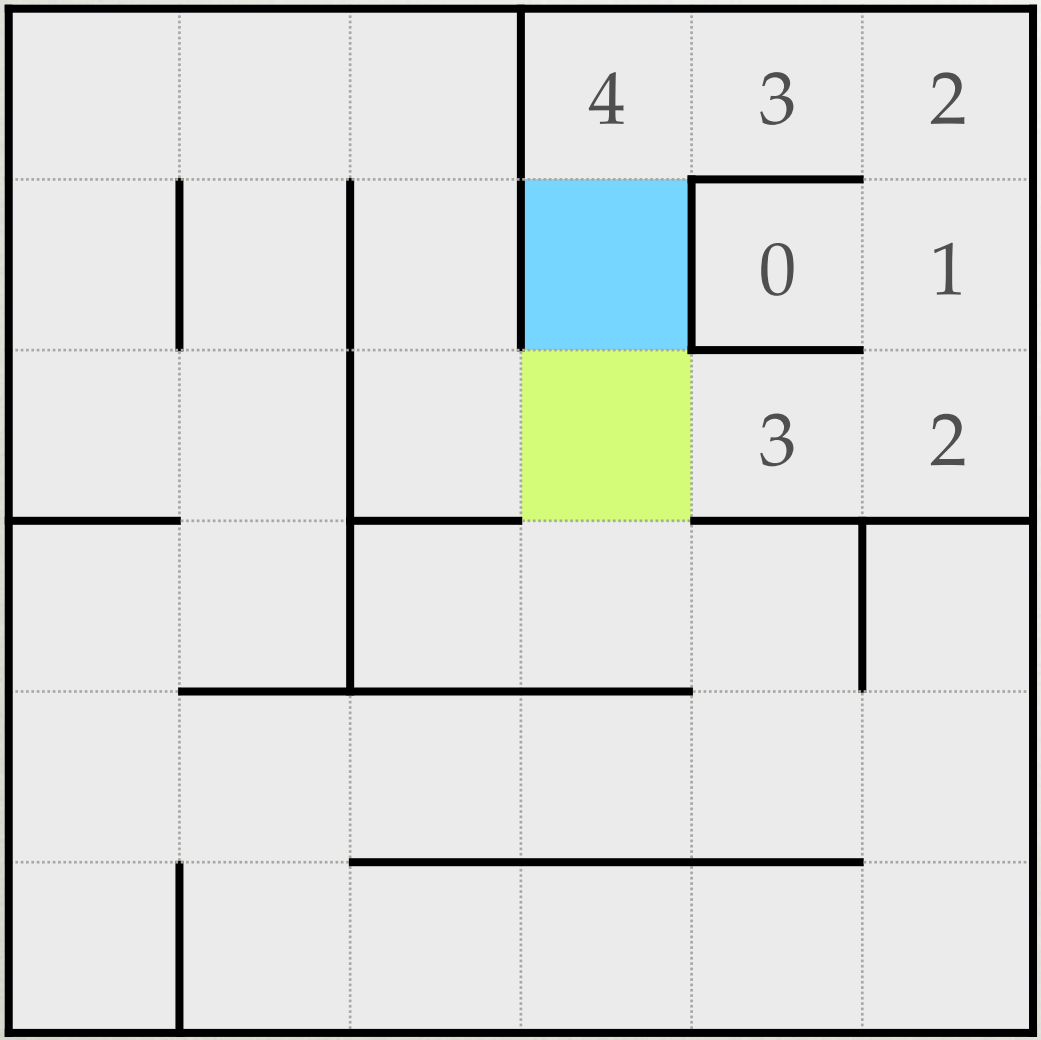
DISTANCE



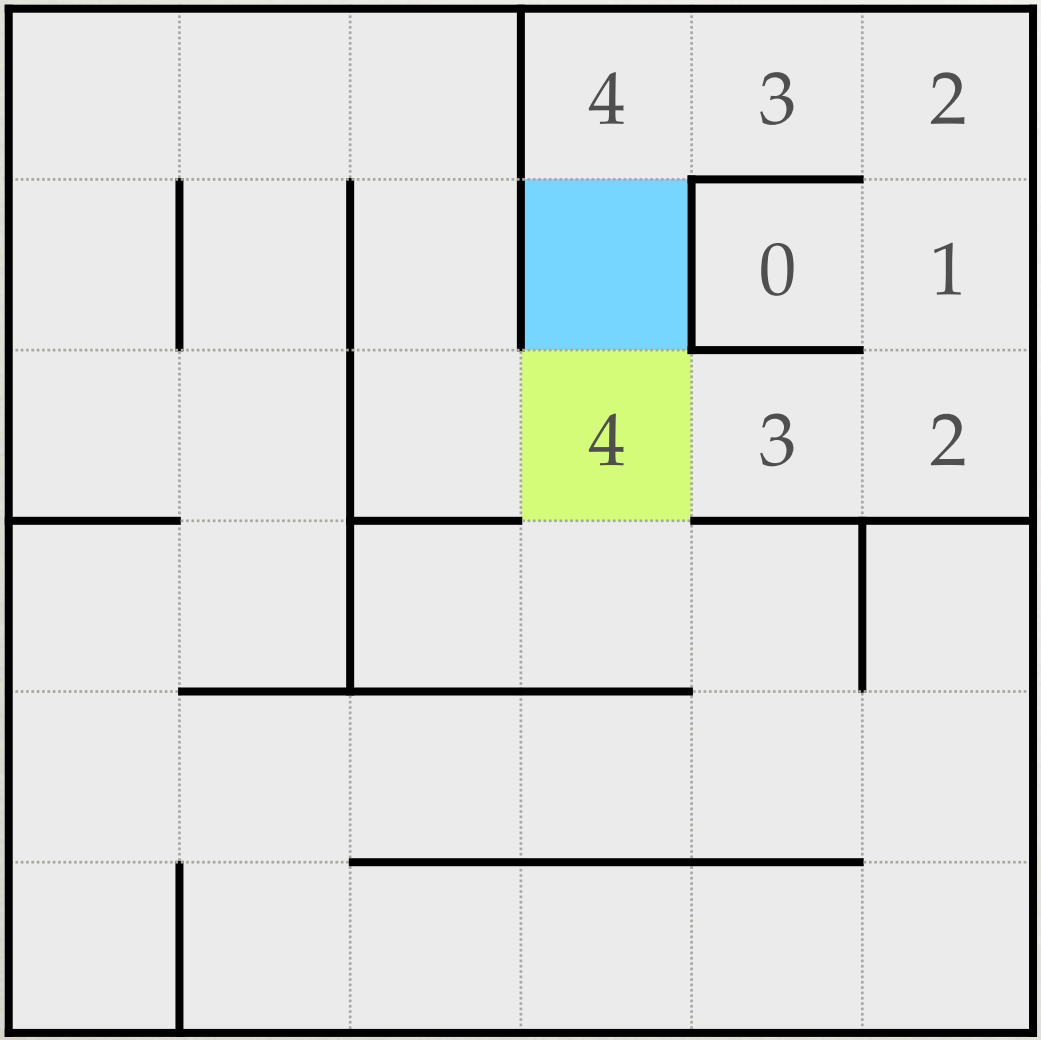
NEIGHBOURS



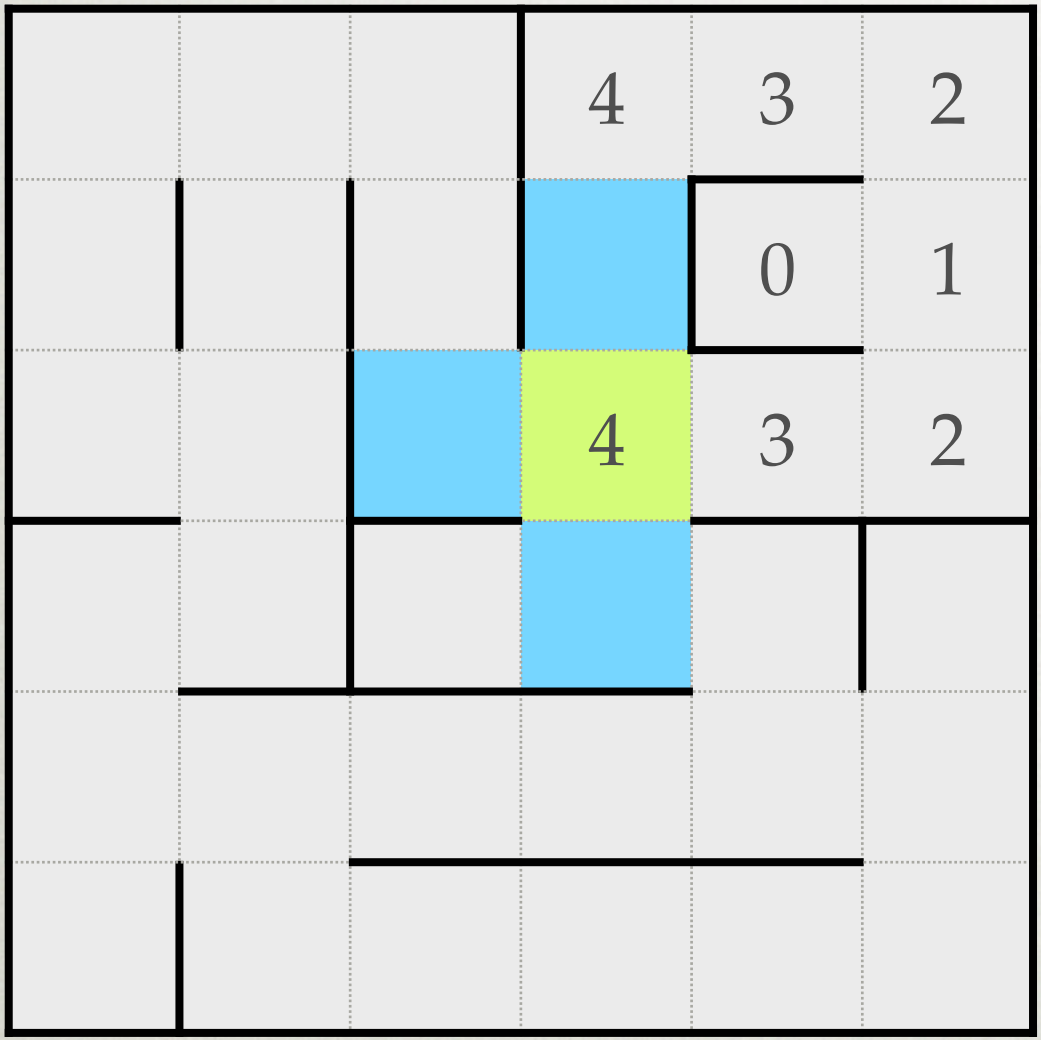
HEAD



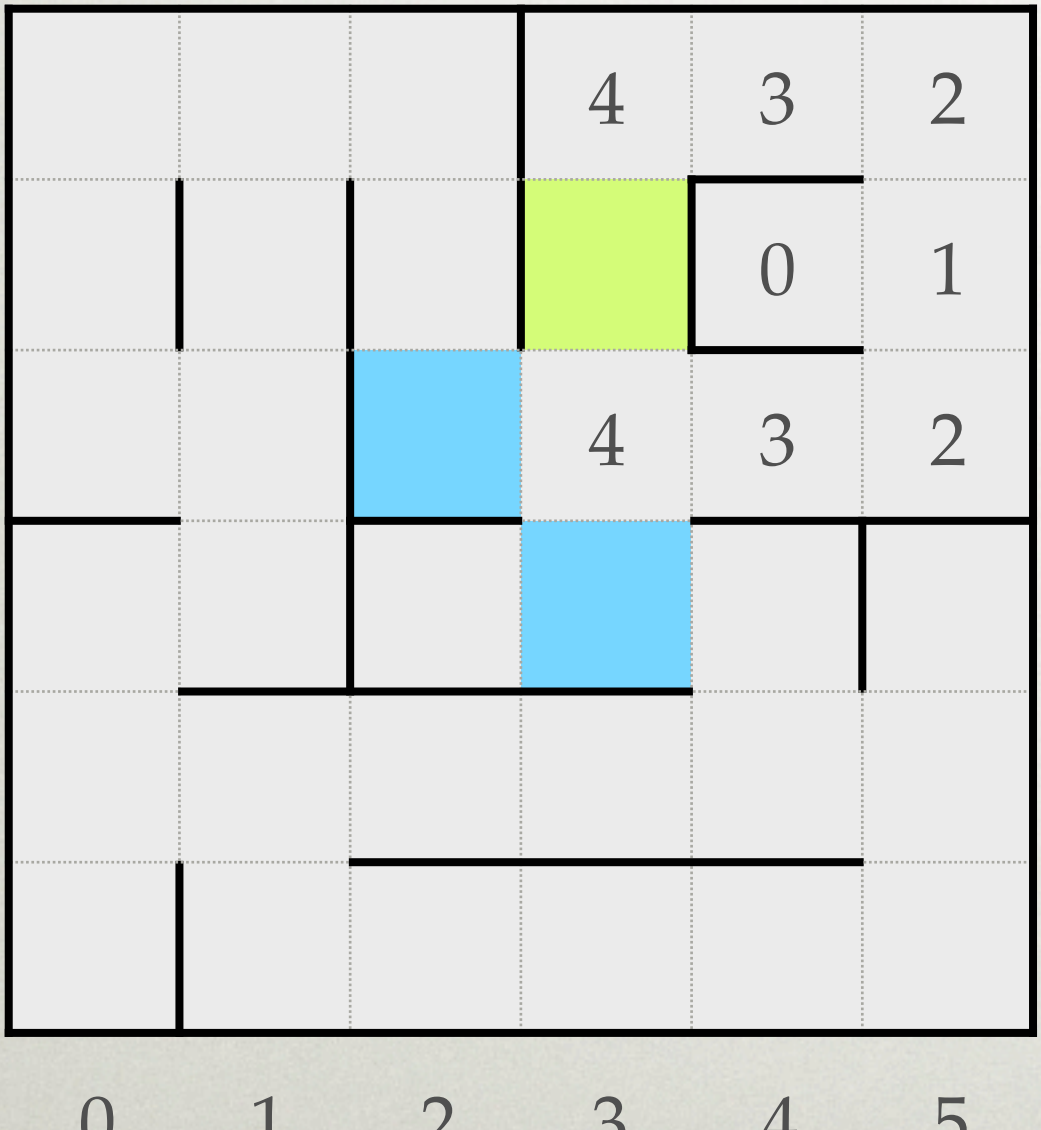
DISTANCE



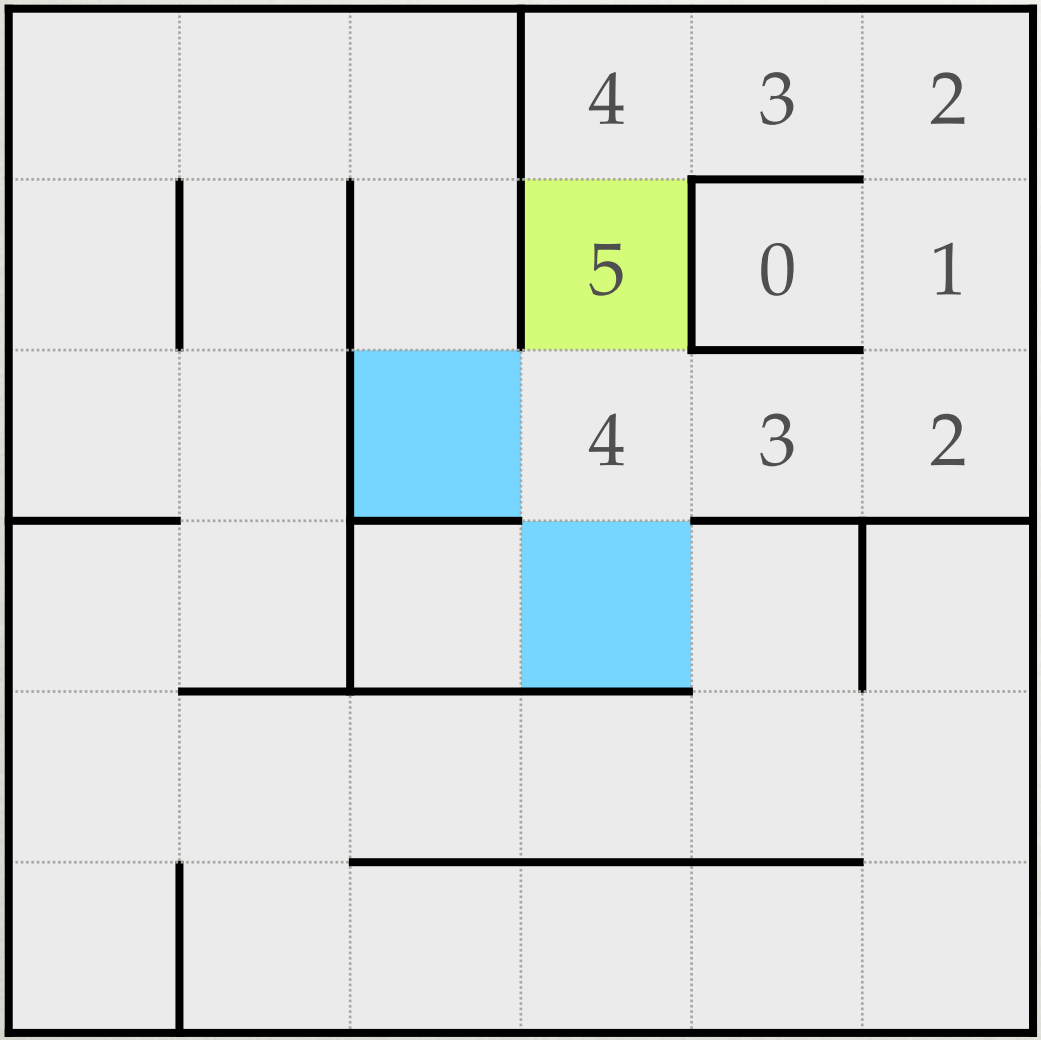
NEIGHBOURS



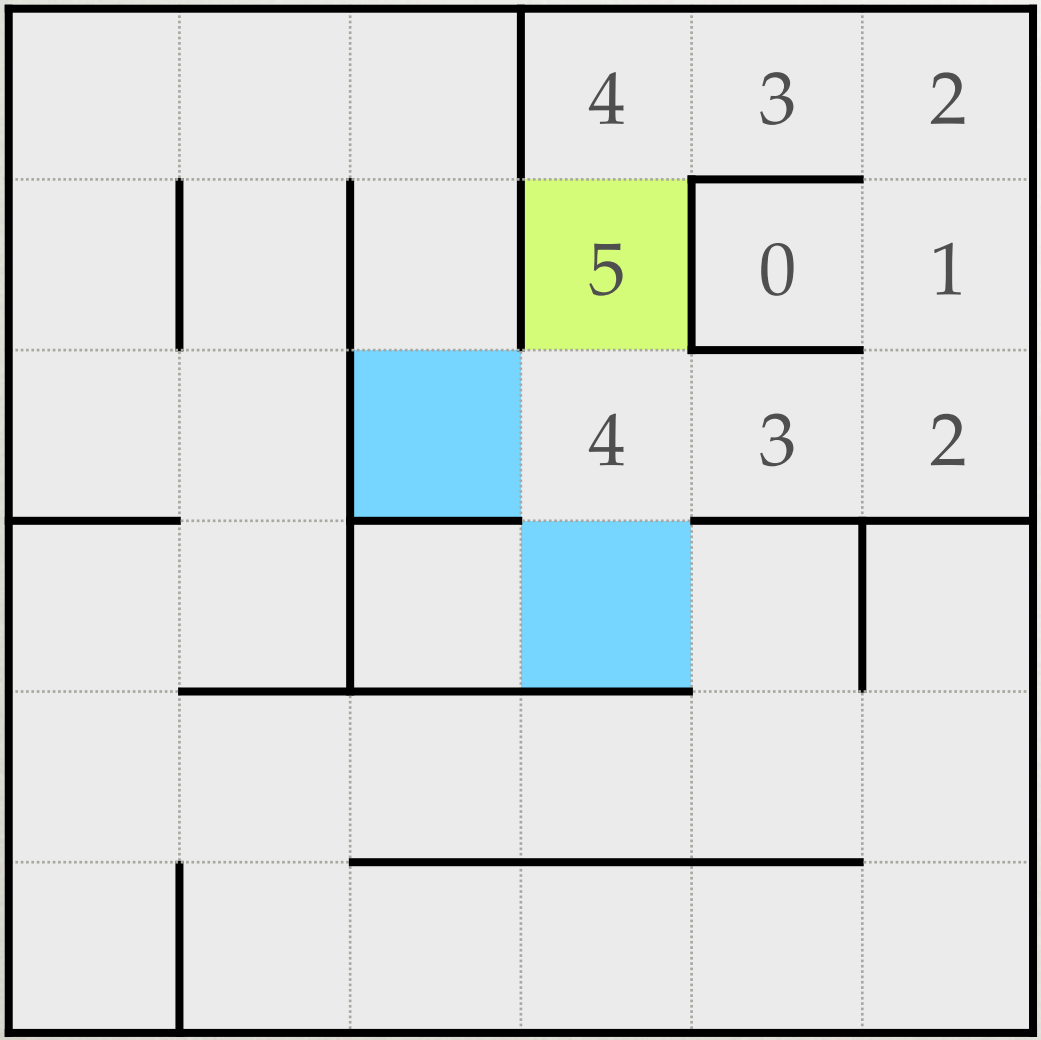
HEAD



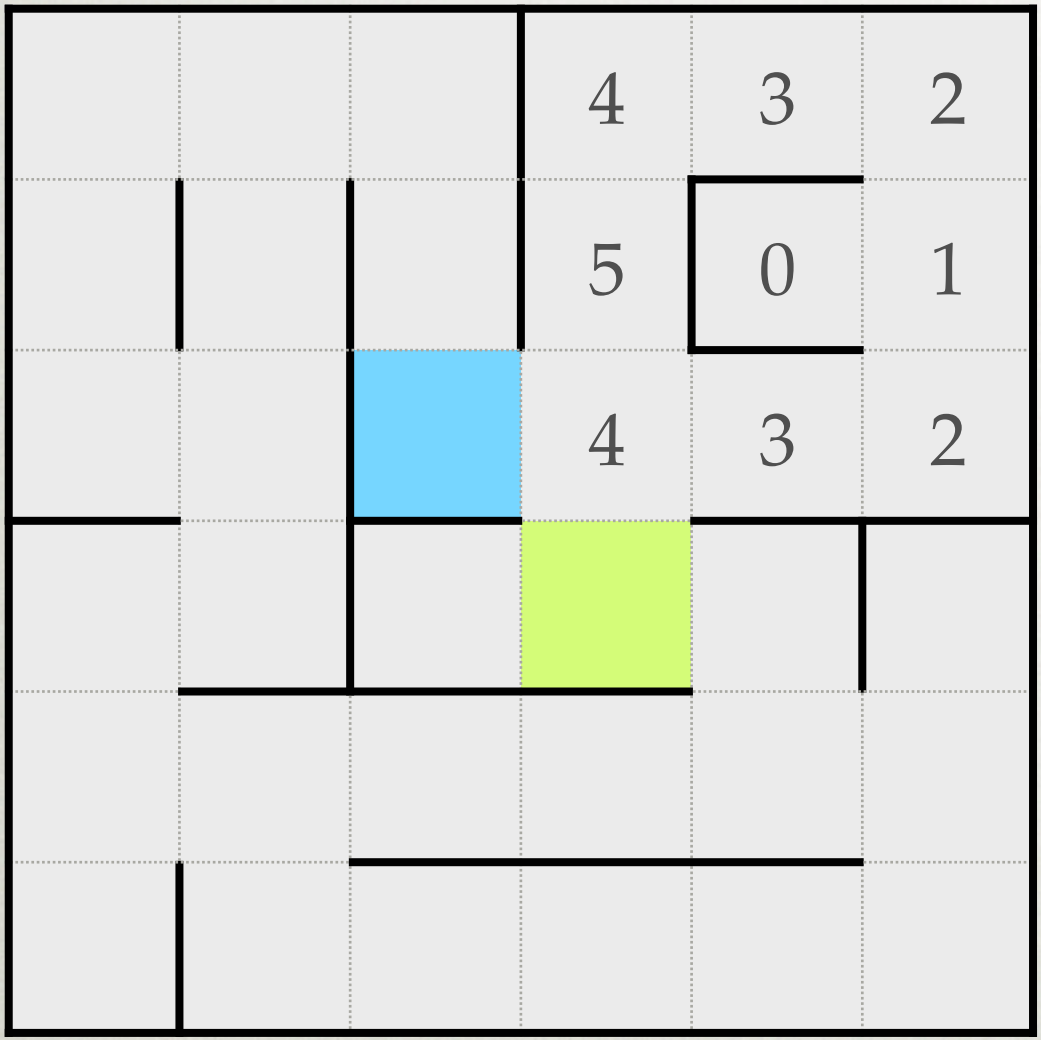
DISTANCE



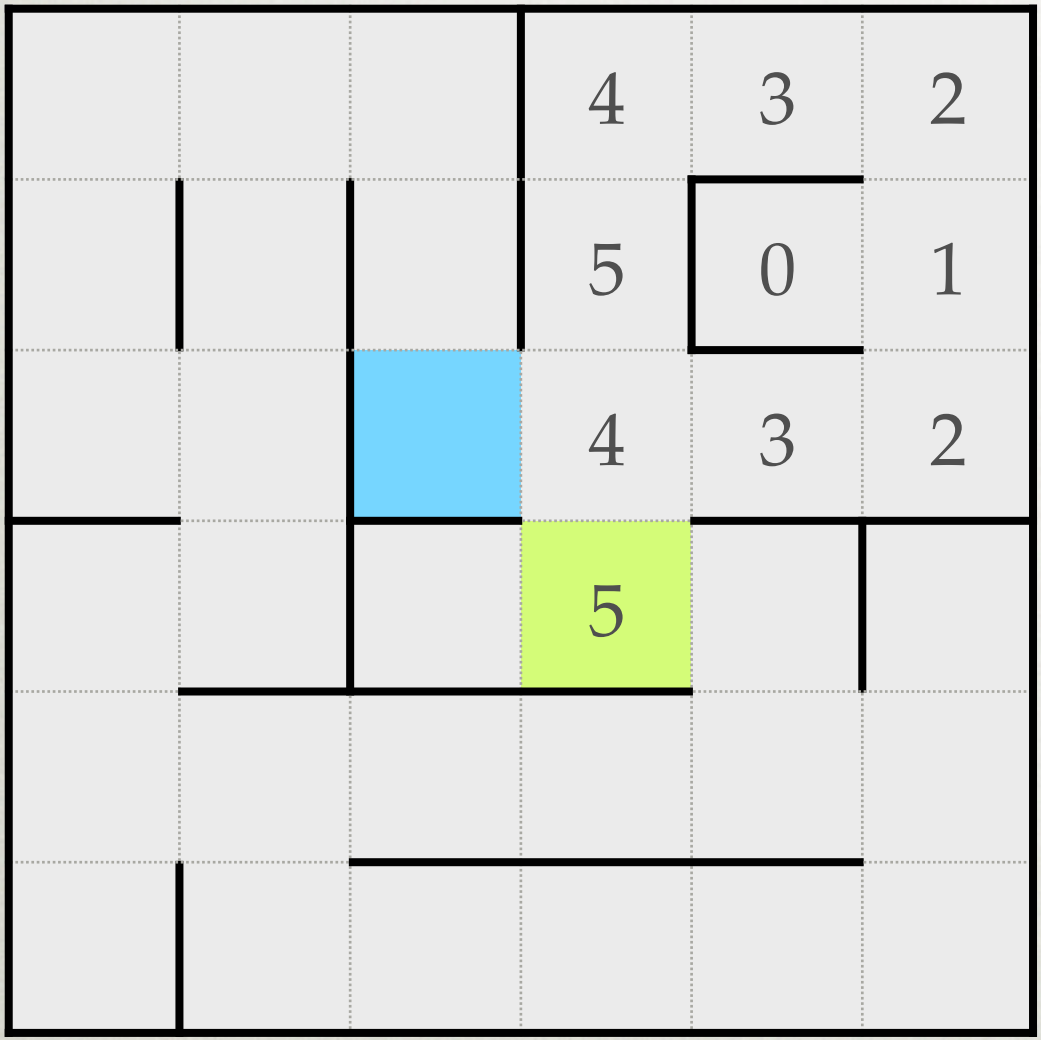
NEIGHBOURS



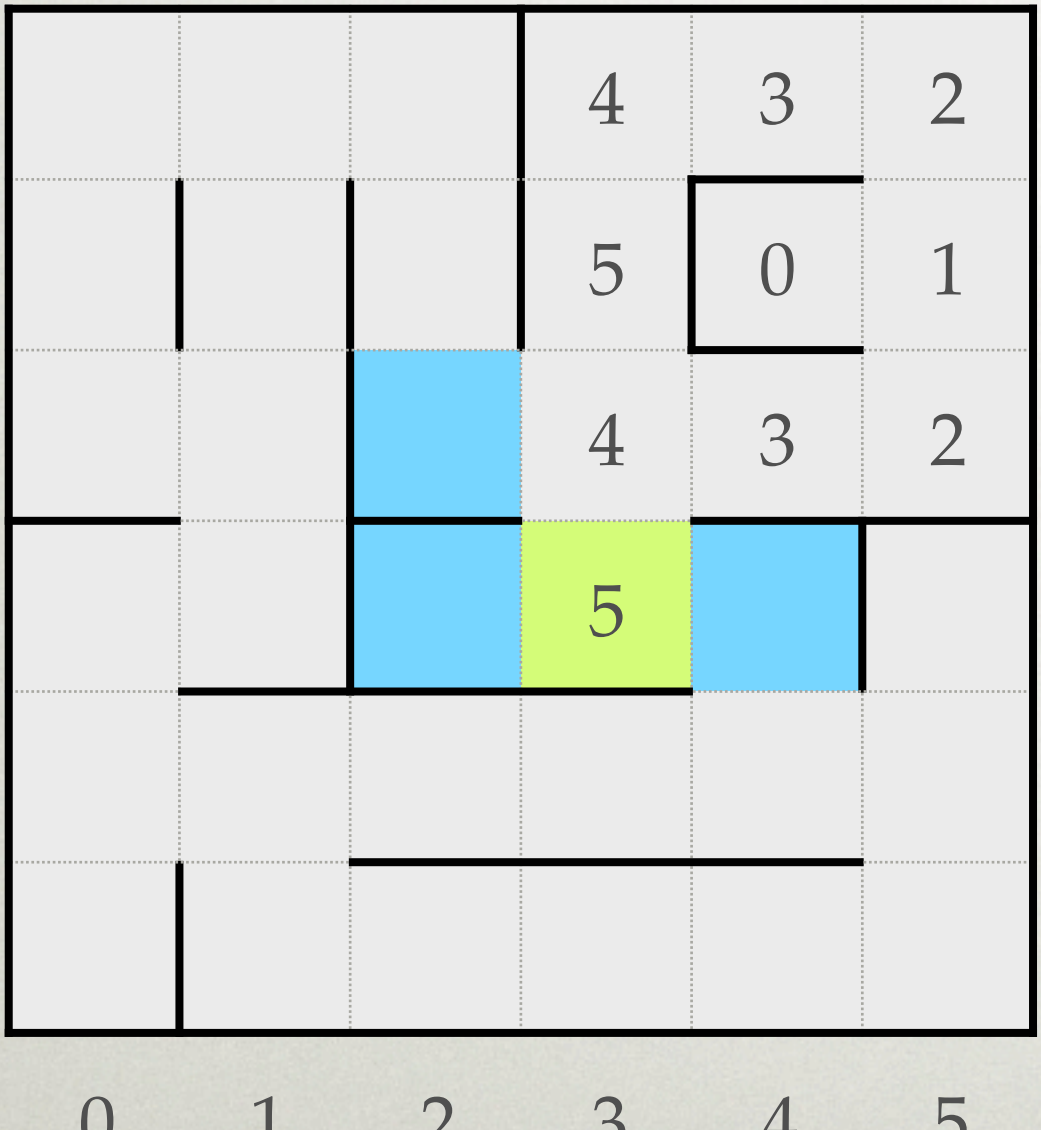
HEAD



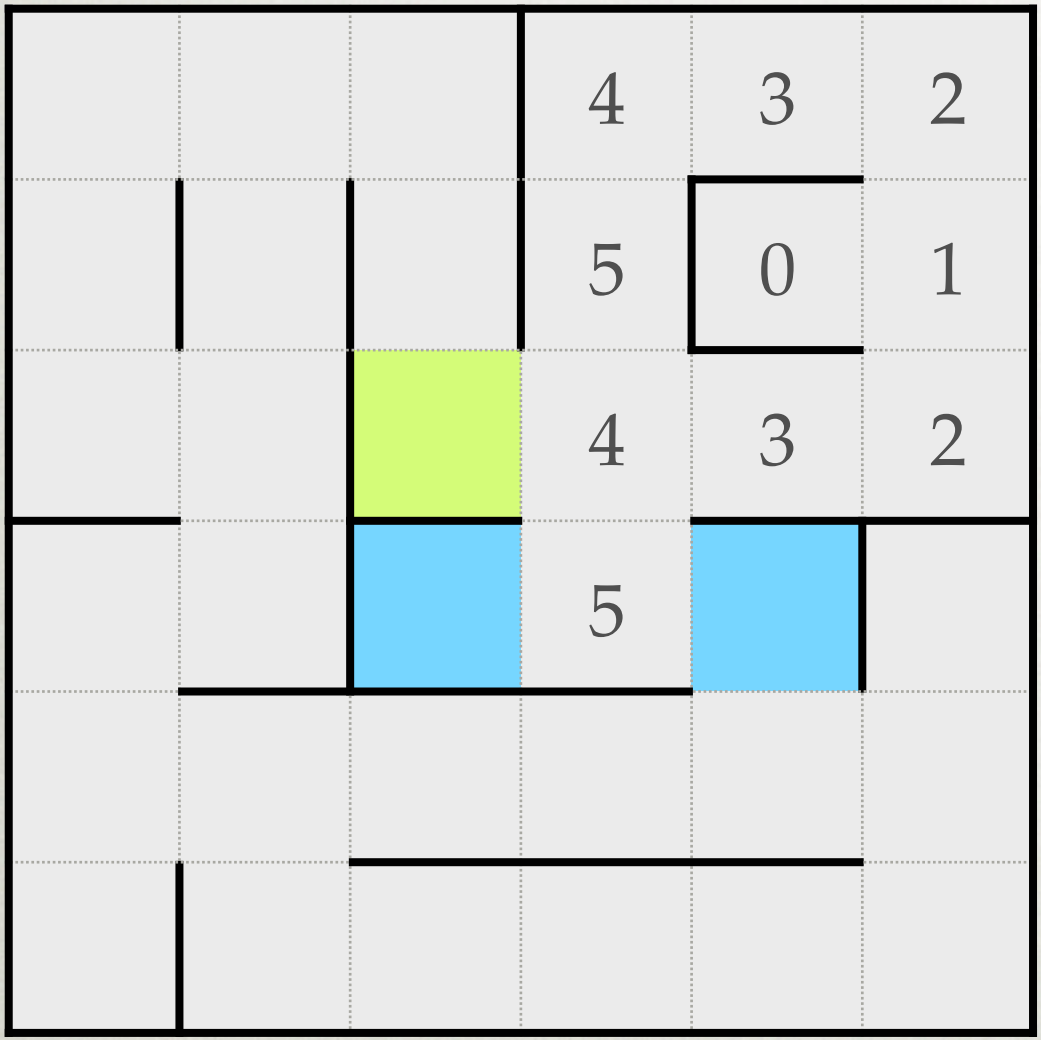
DISTANCE



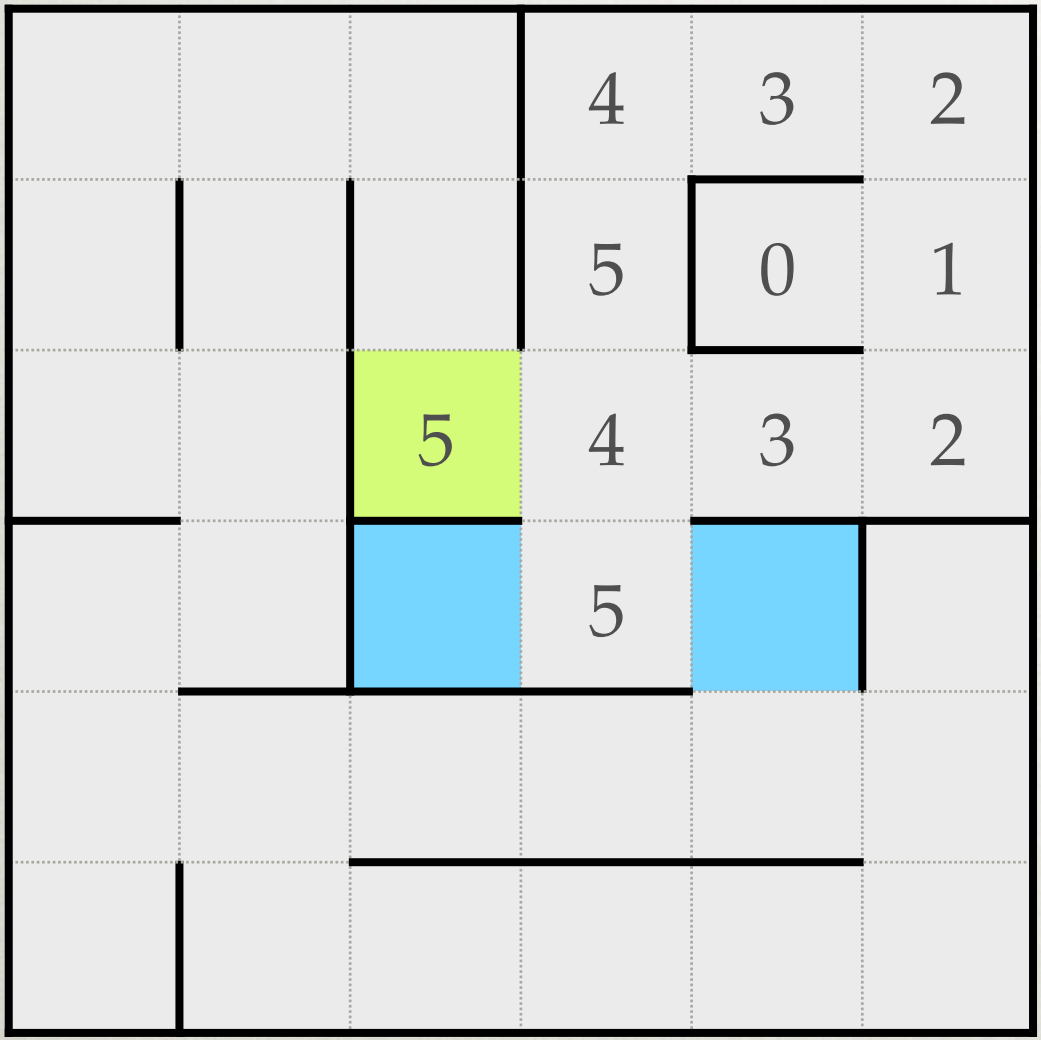
NEIGHBOURS



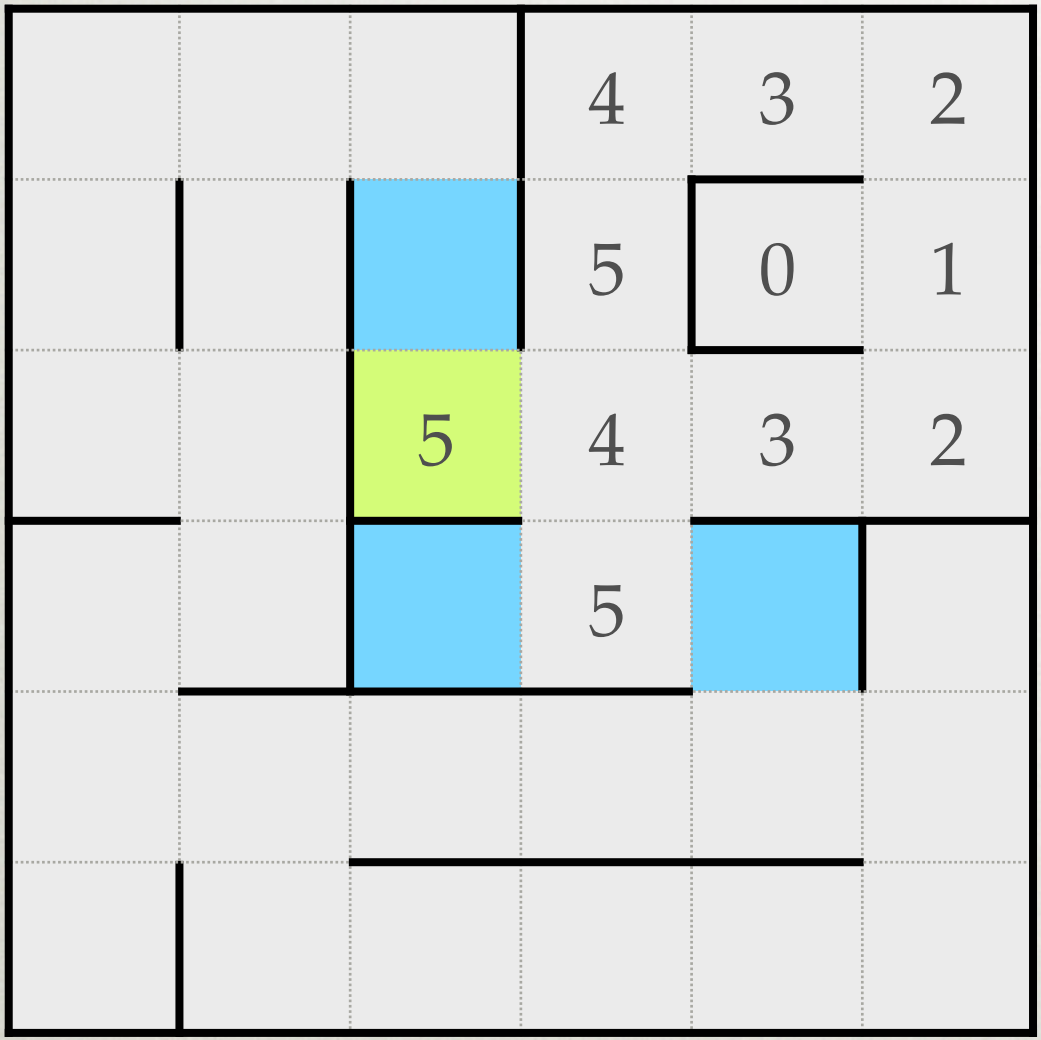
HEAD



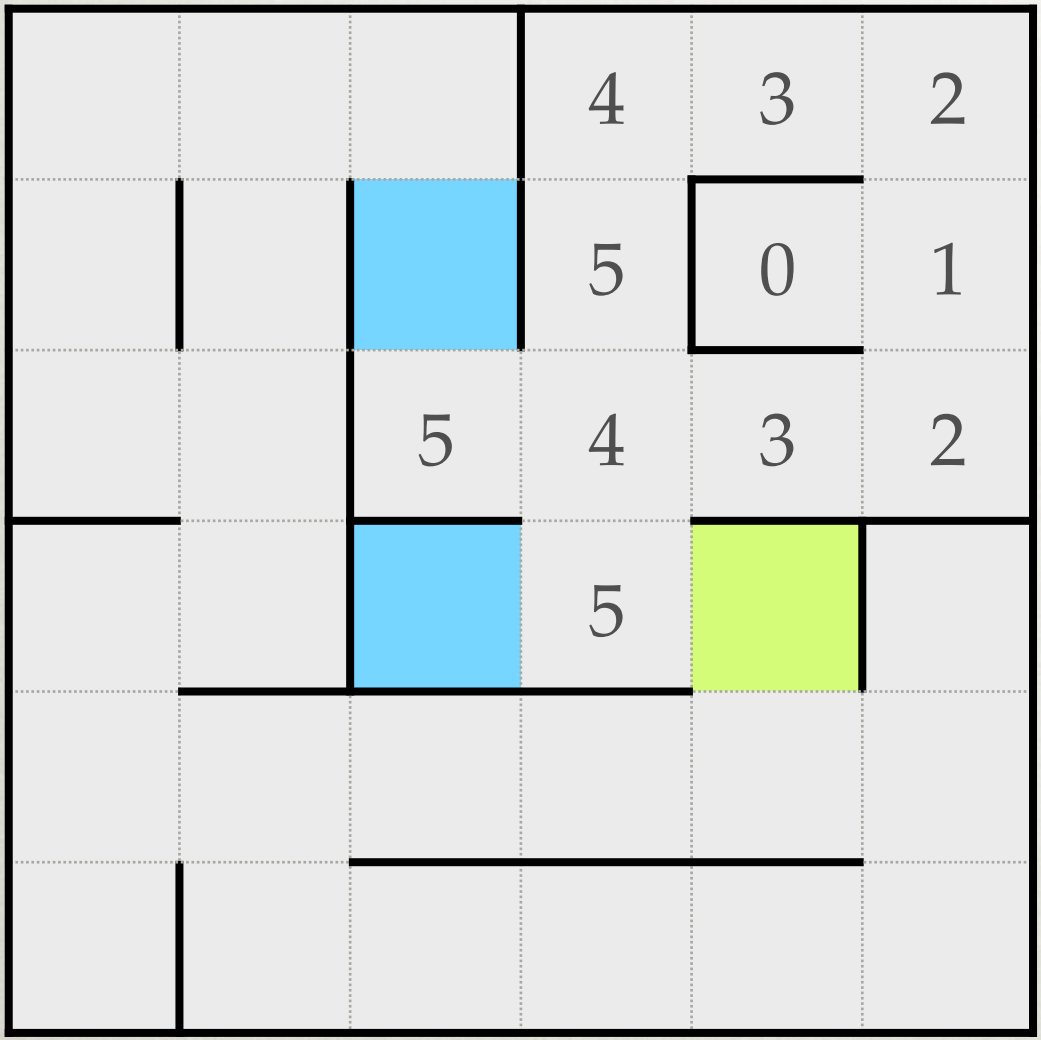
DISTANCE



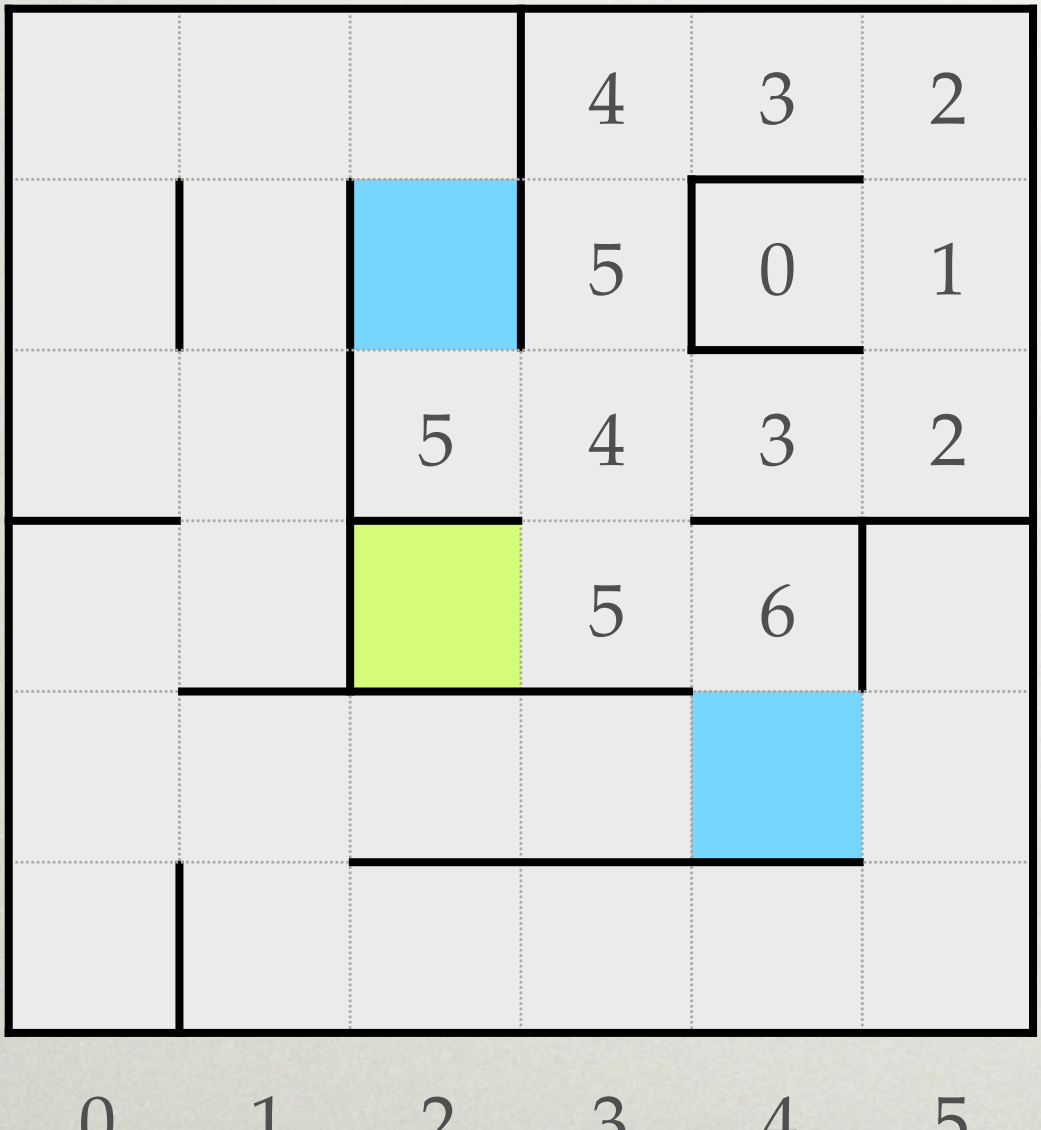
NEIGHBOURS

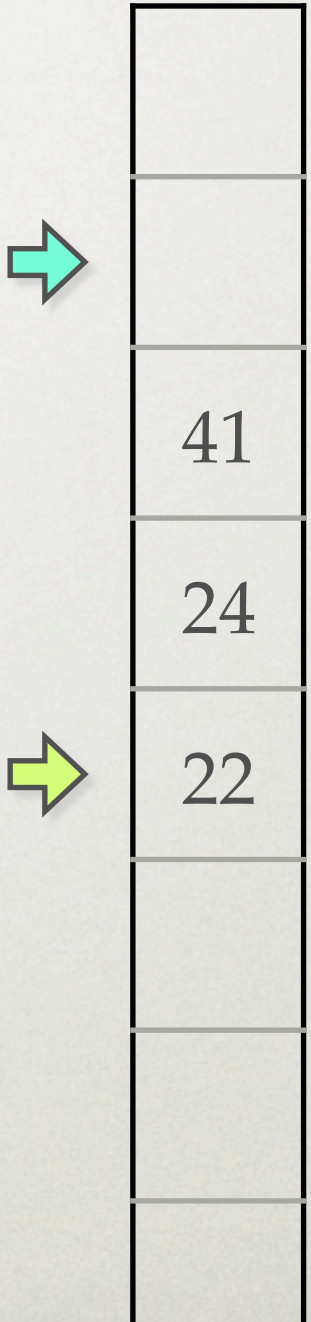


HEAD

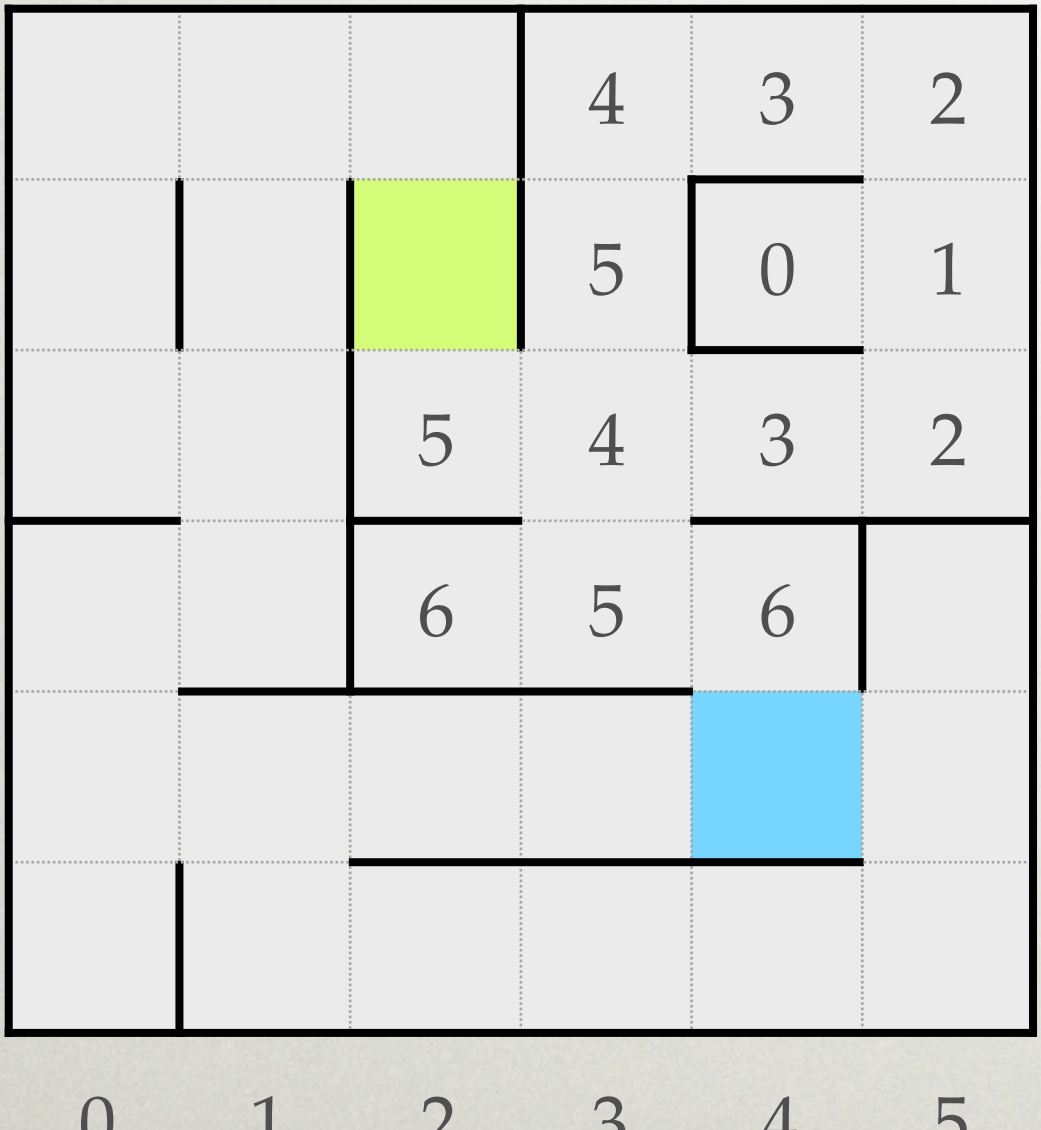


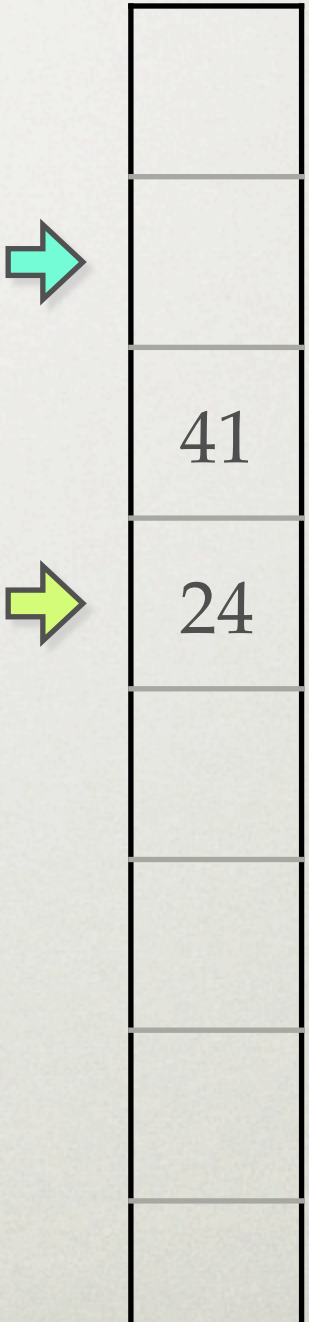
FULL STEP



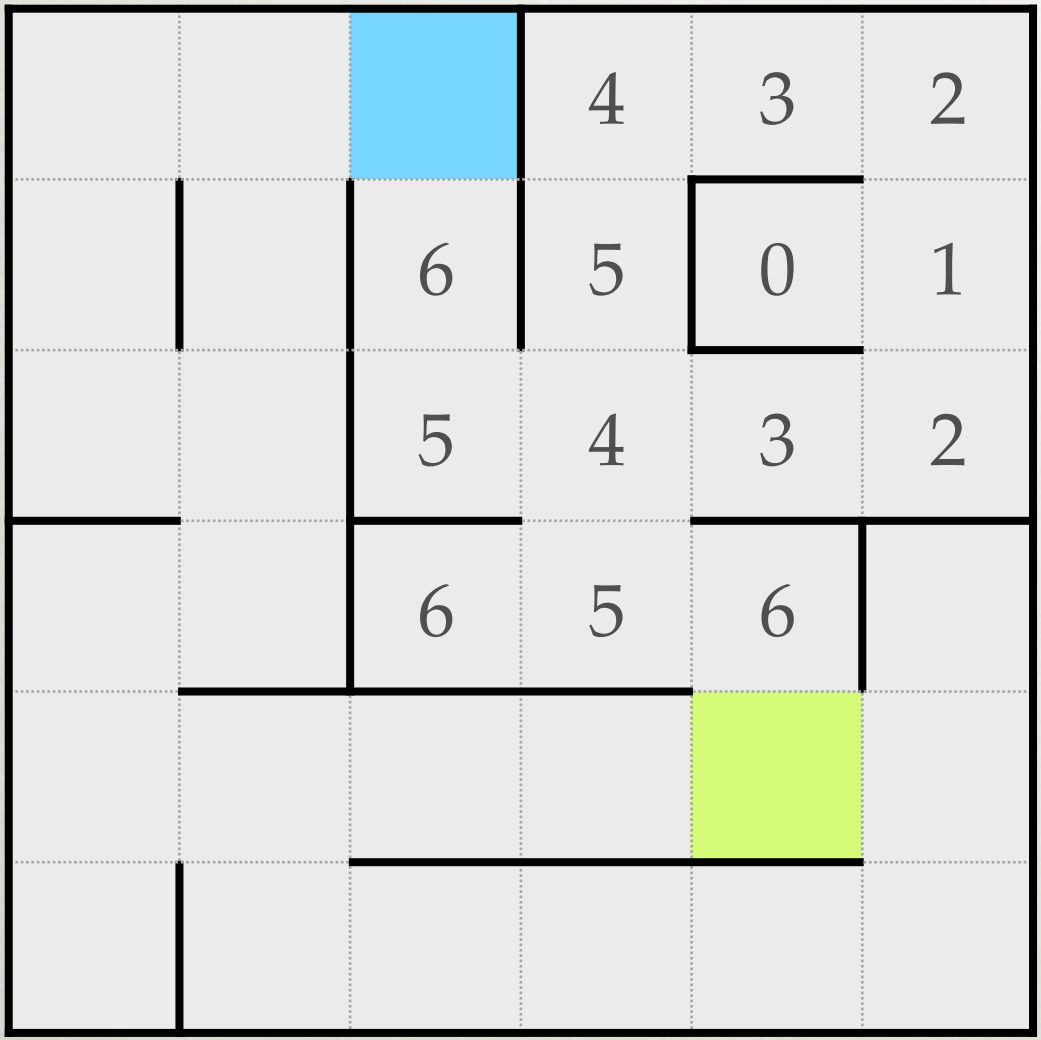


FULL STEP

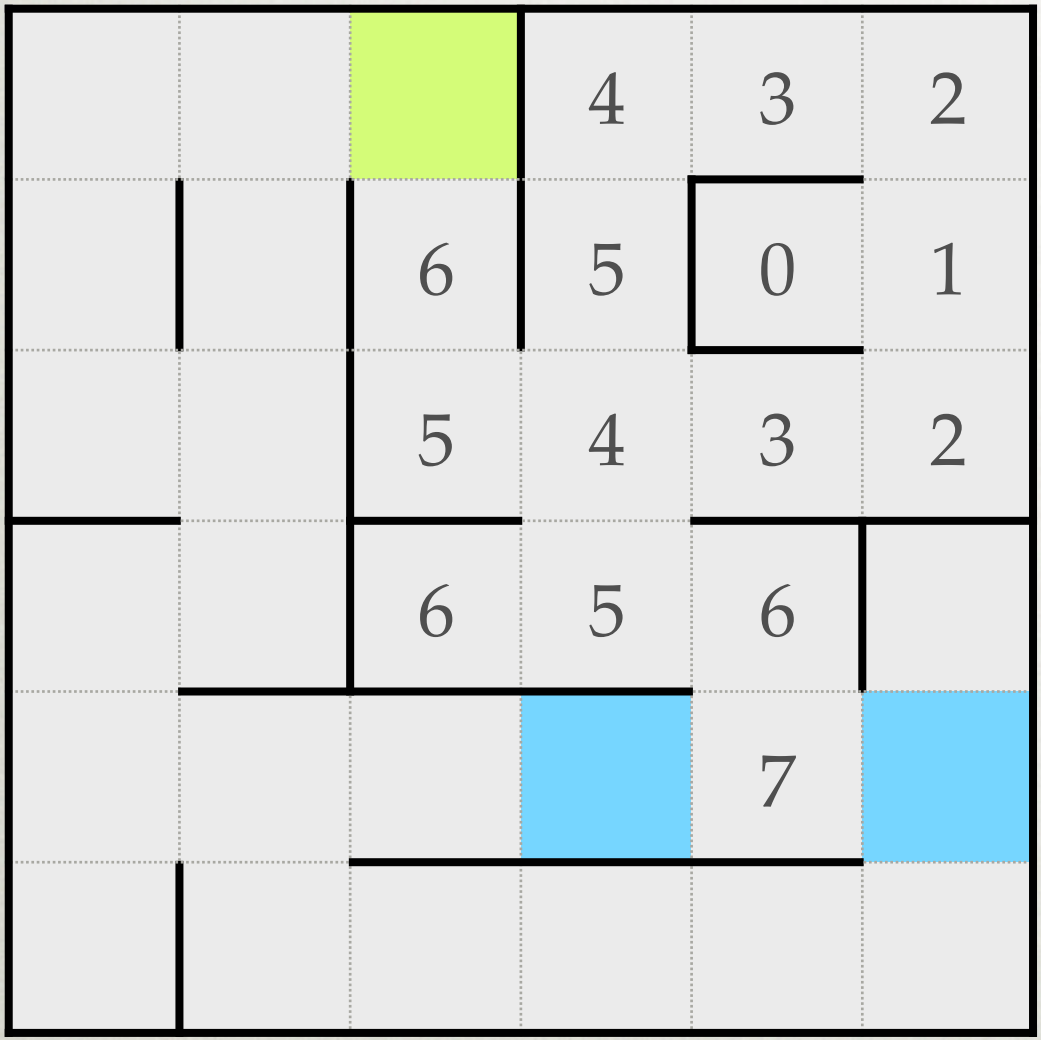




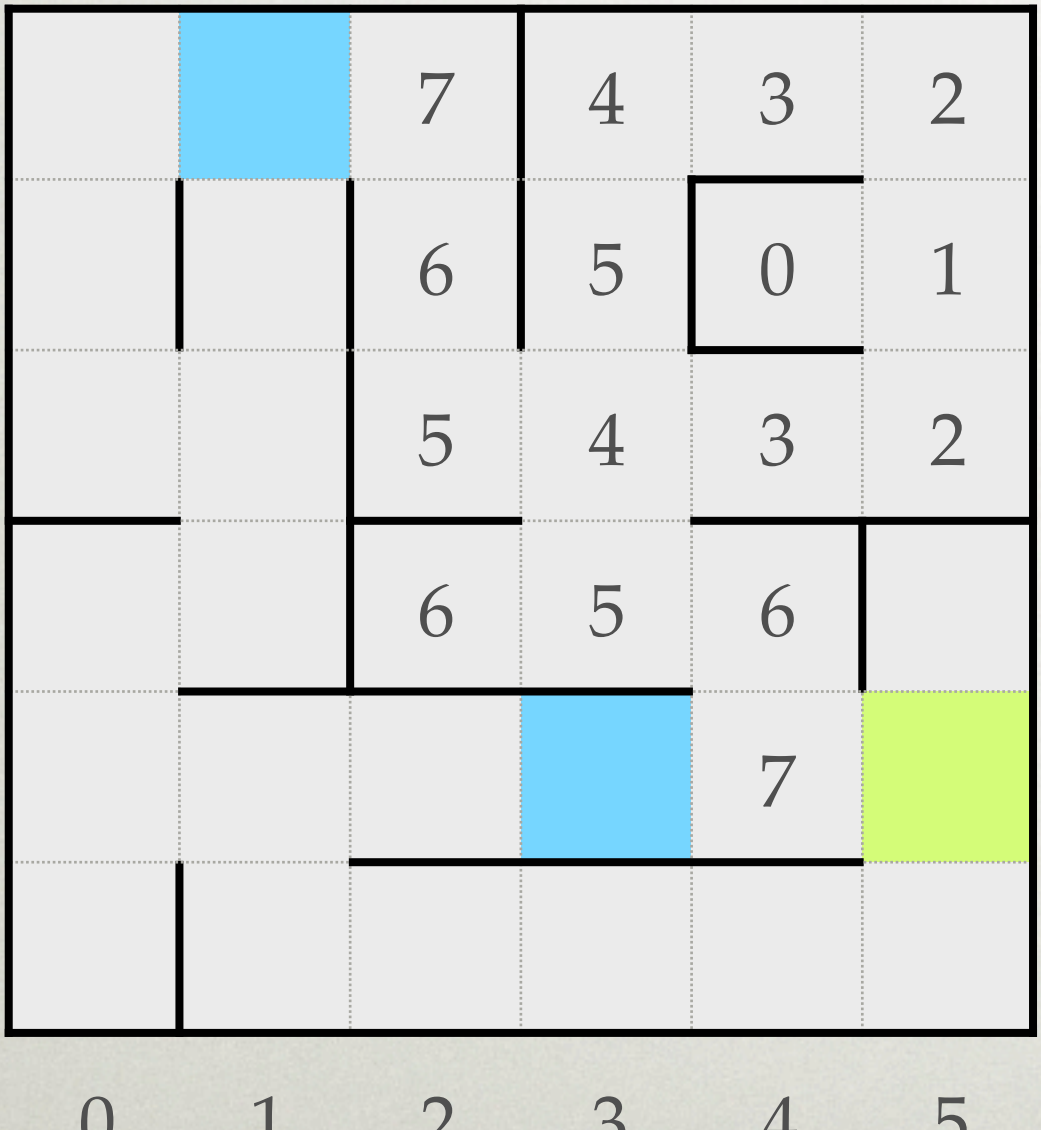
FULL STEP



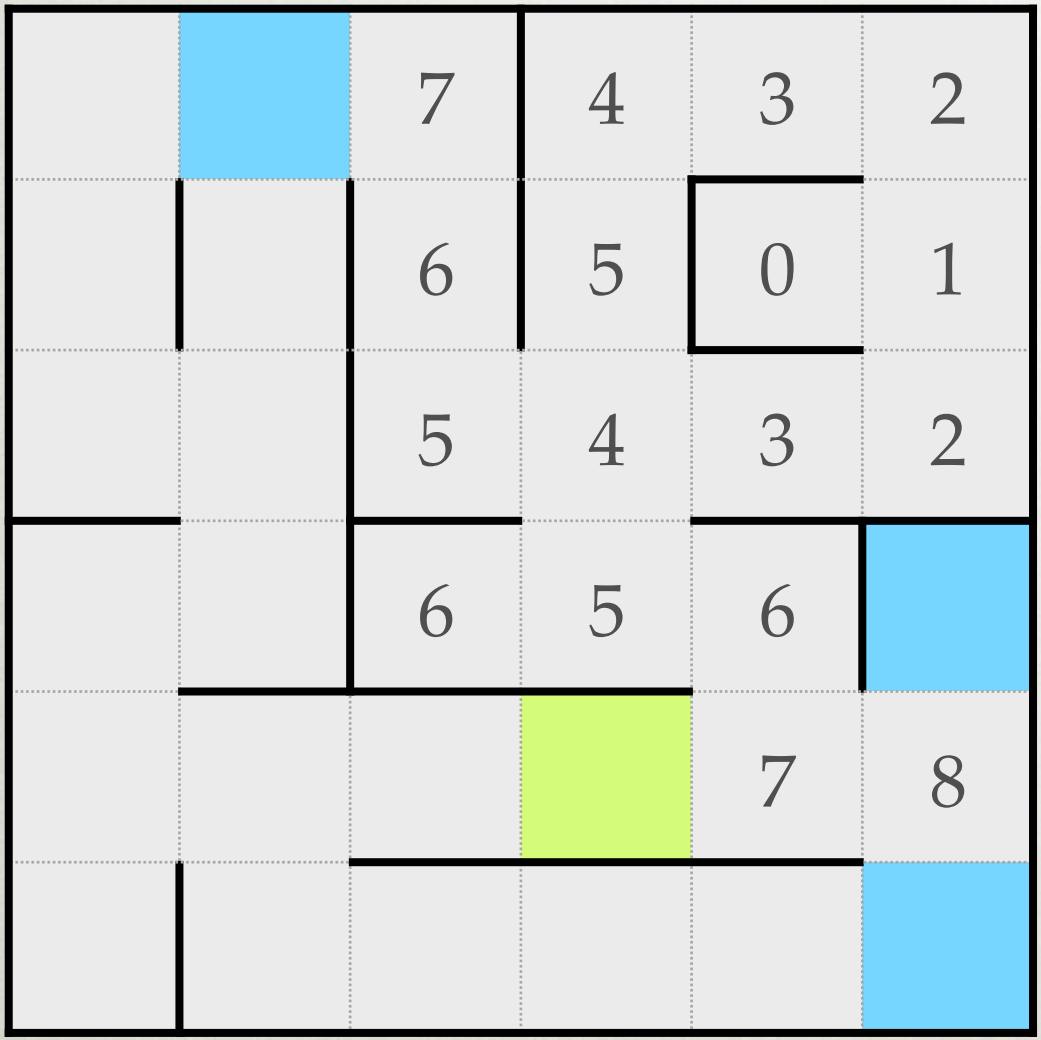
FULL STEP



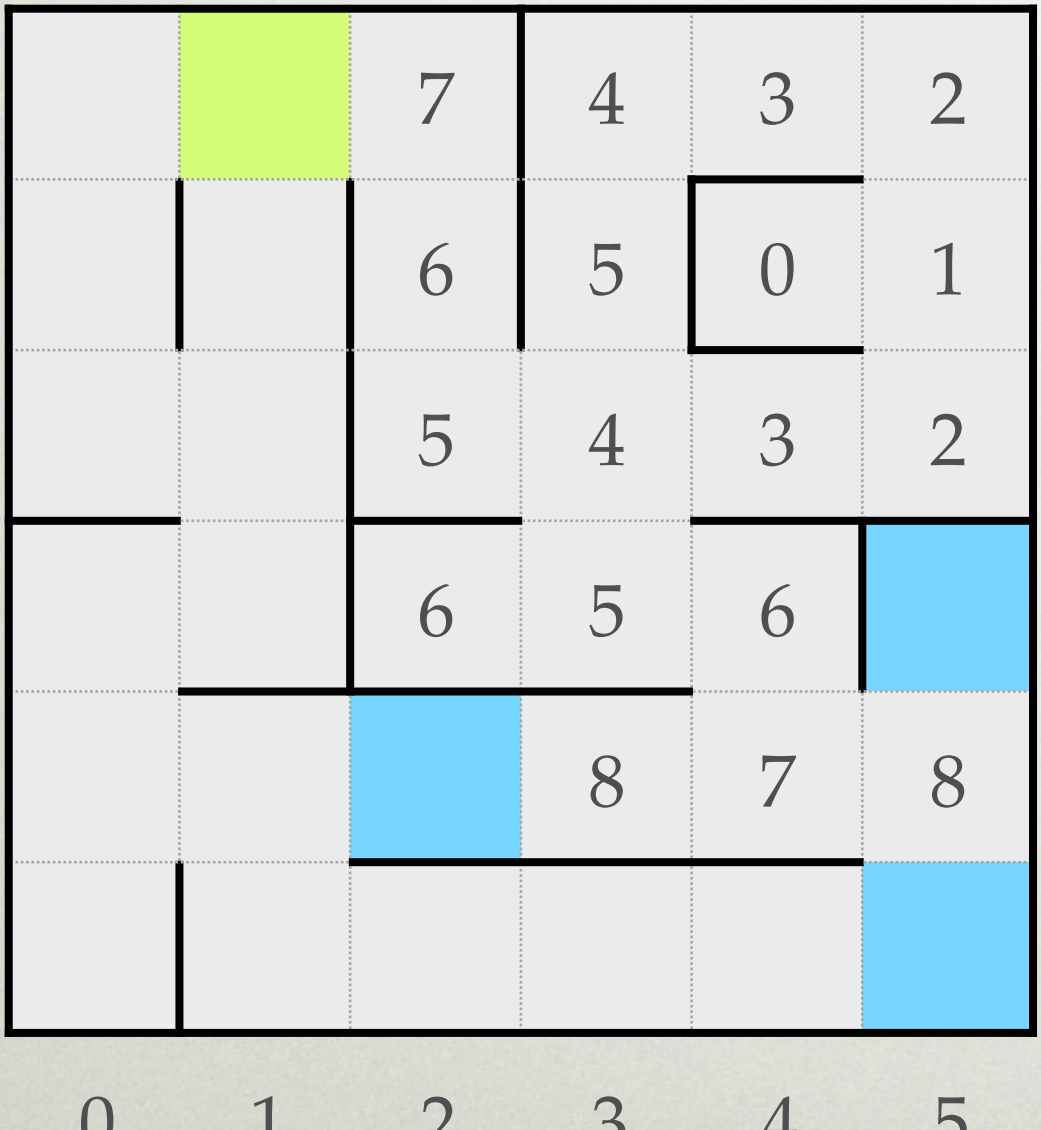
FULL STEP

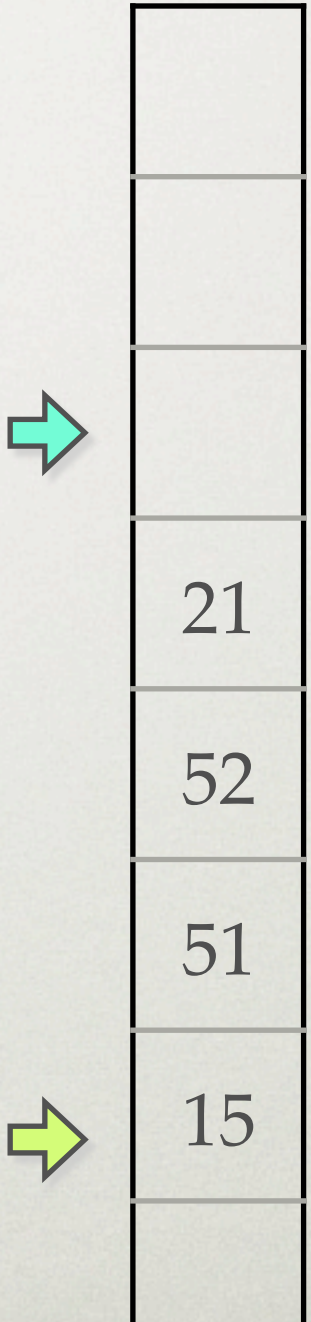


FULL STEP

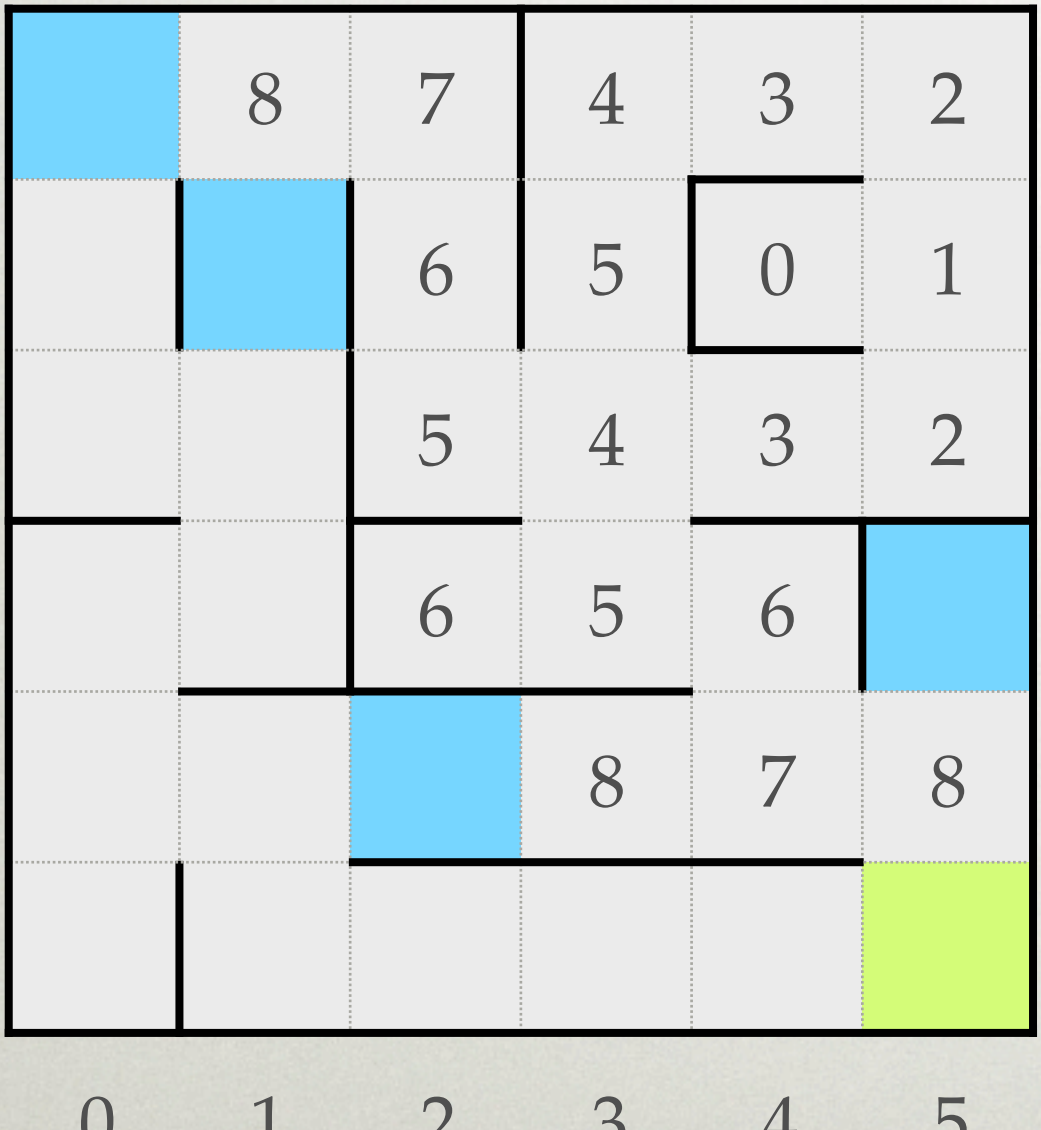


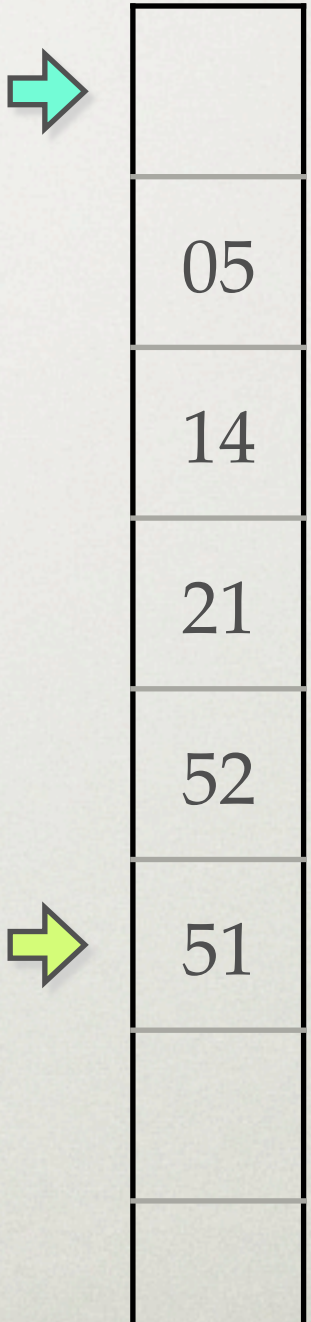
FULL STEP



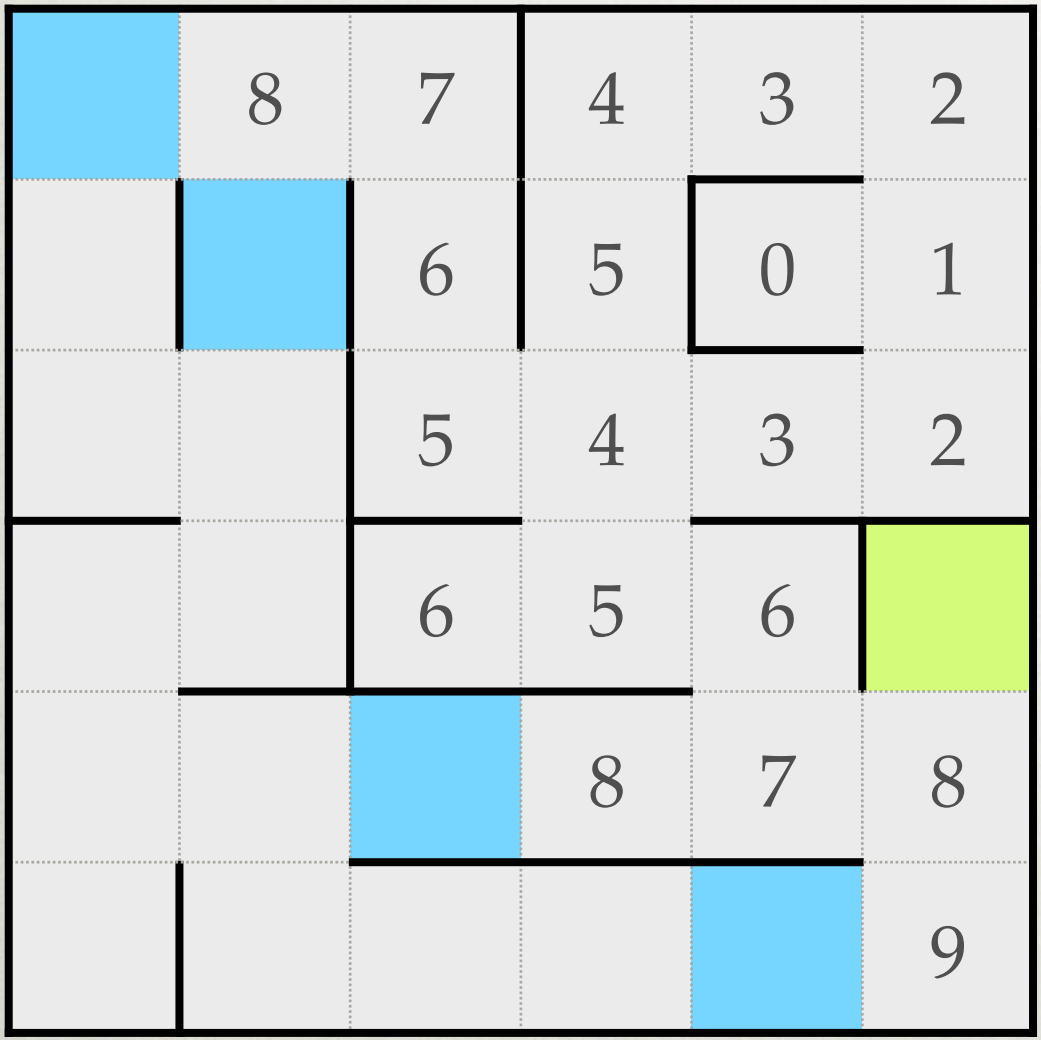


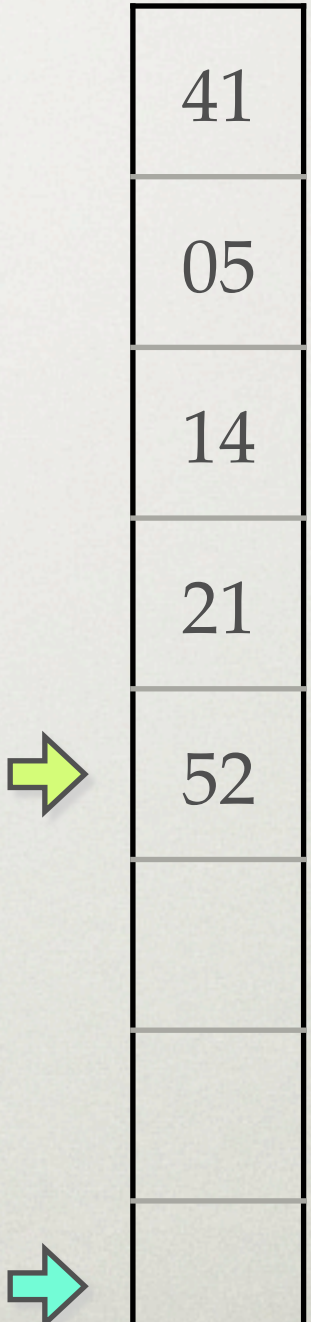
FULL STEP



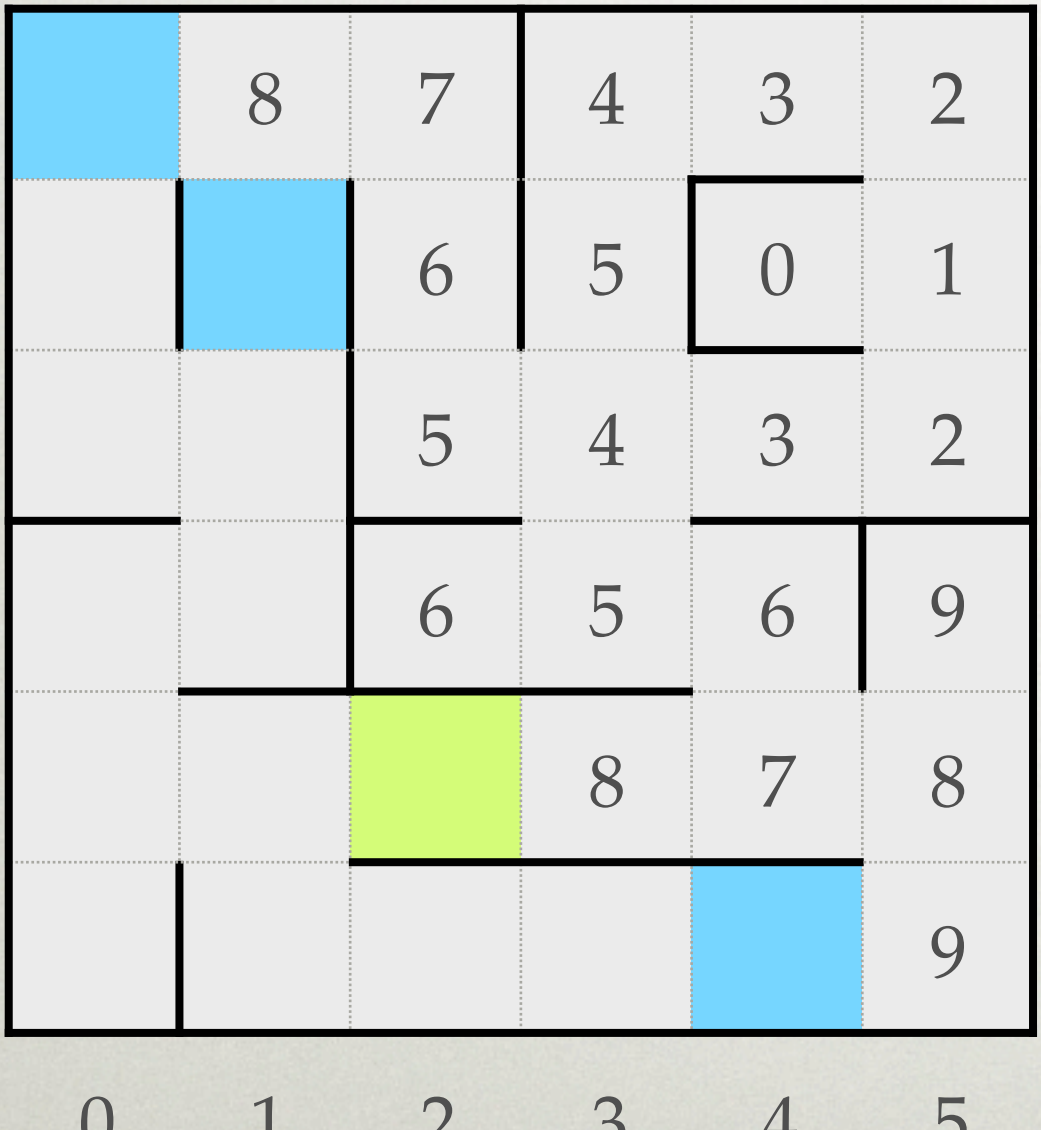


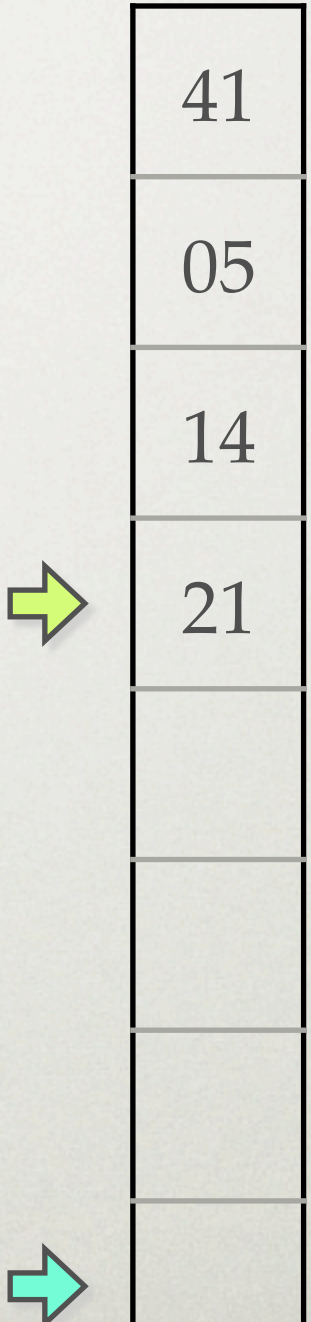
FULL STEP



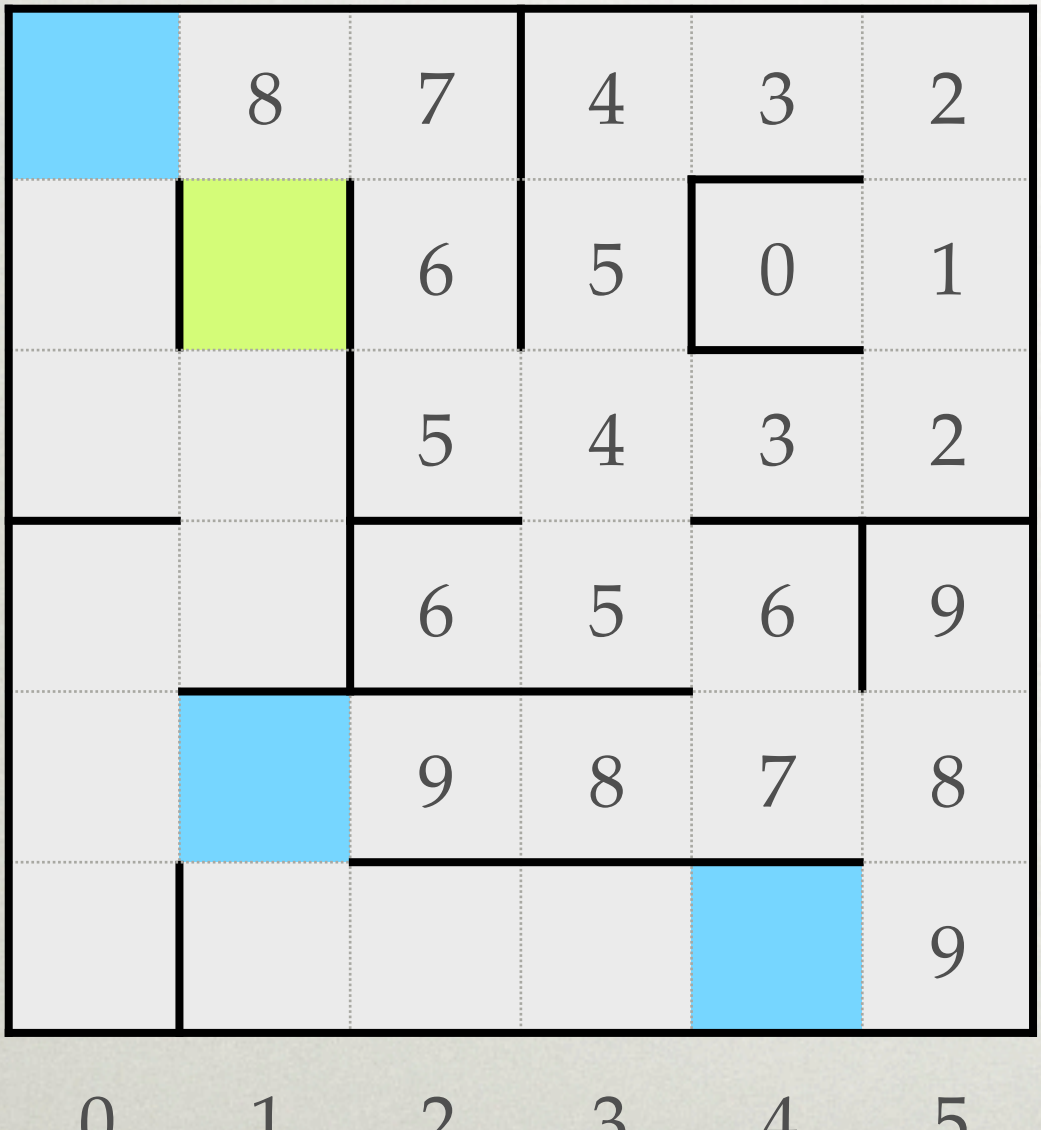


FULL STEP

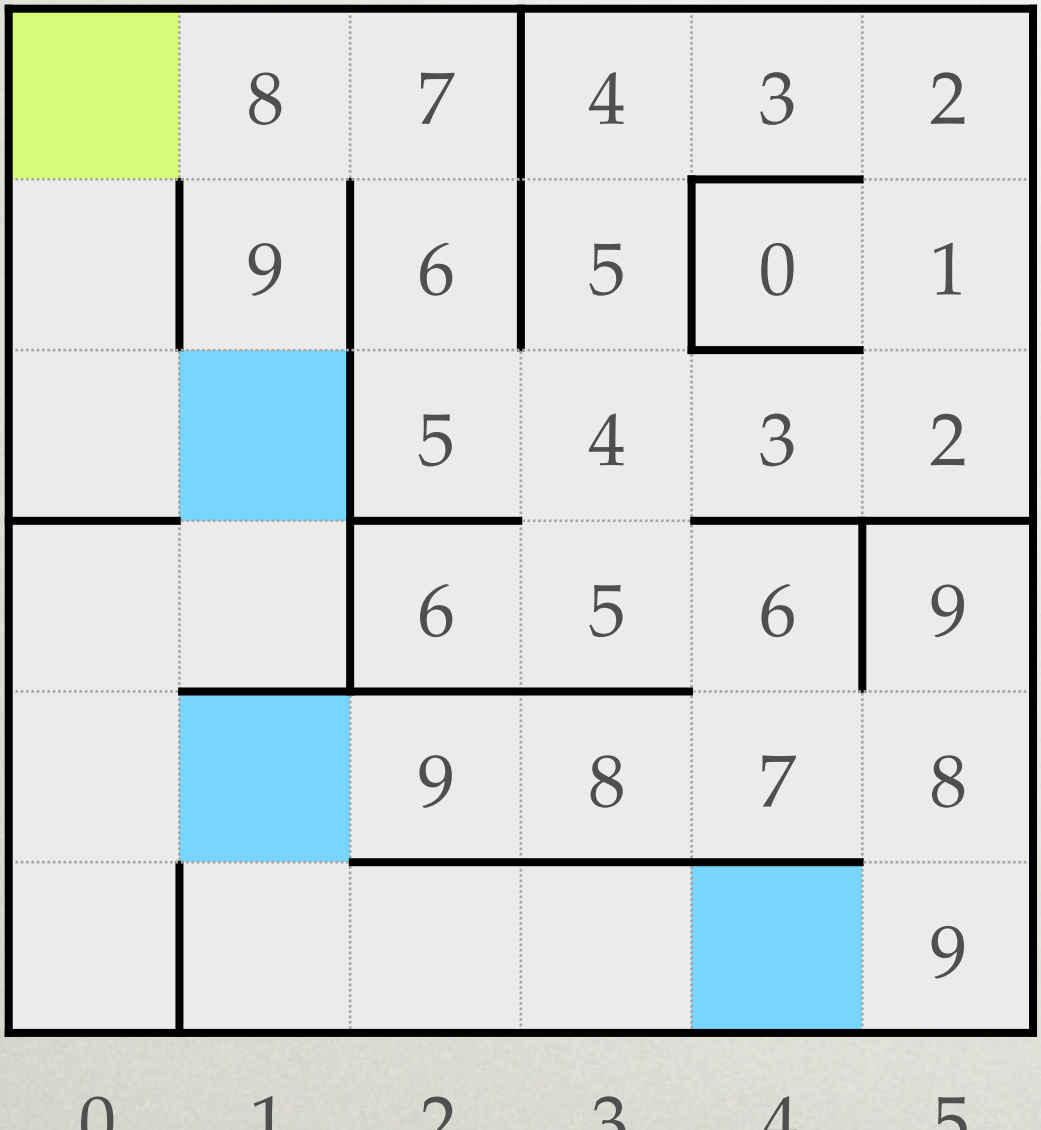




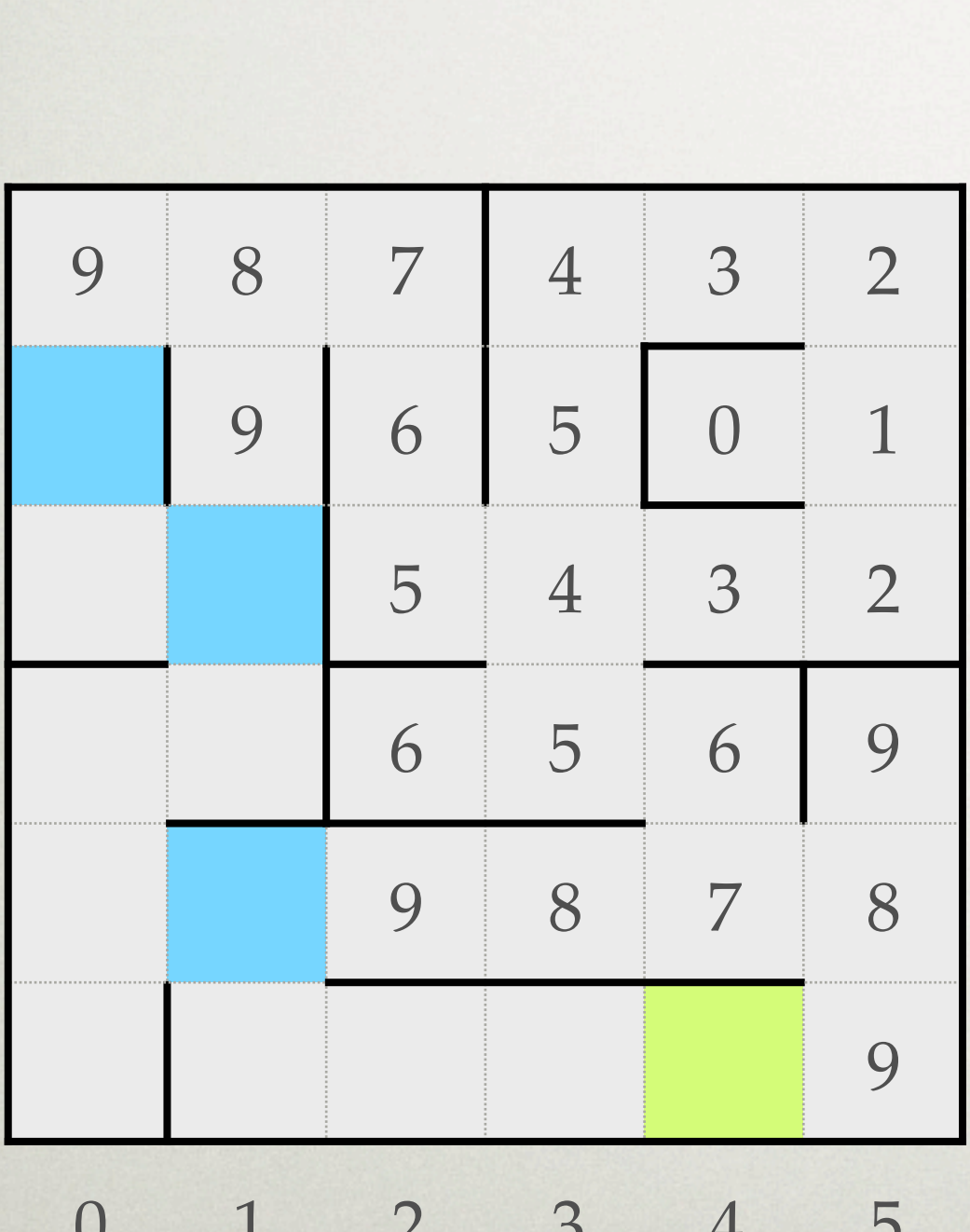
FULL STEP

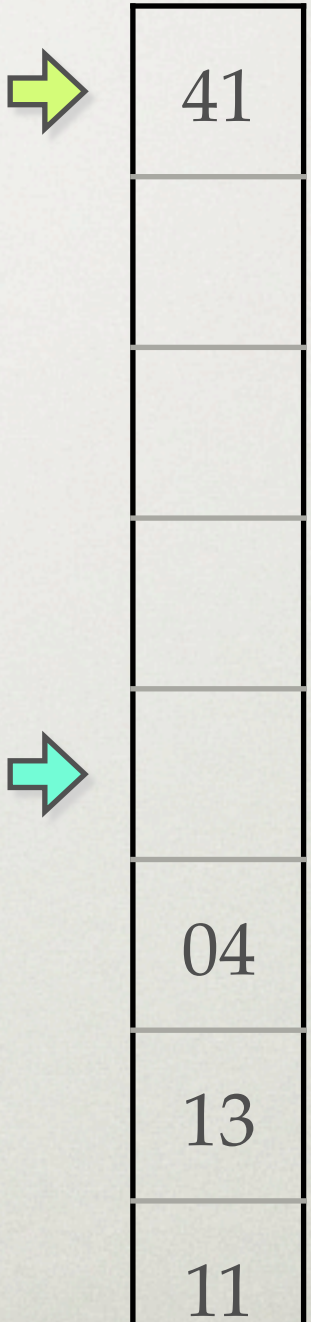


FULL STEP

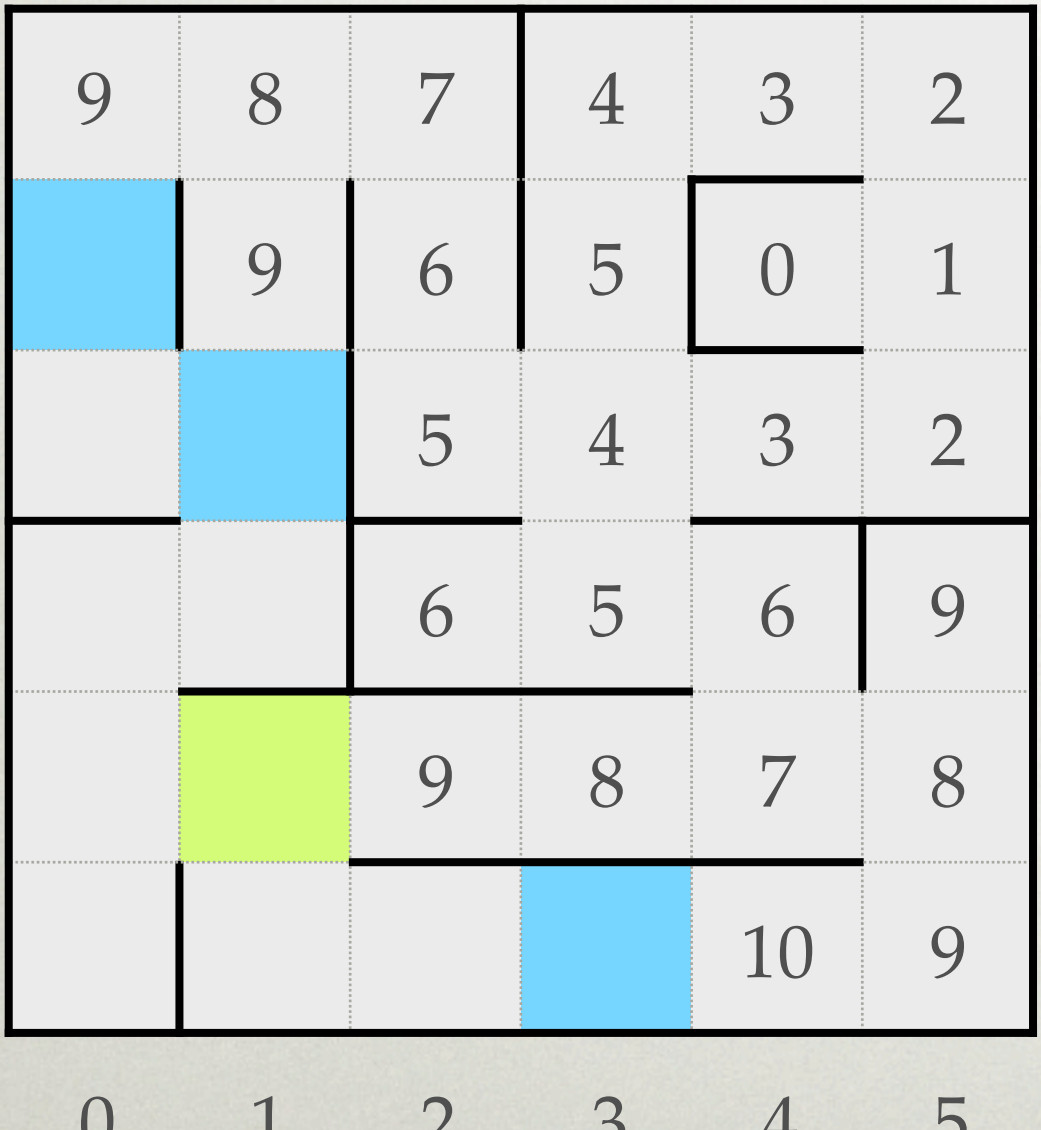


FULL STEP

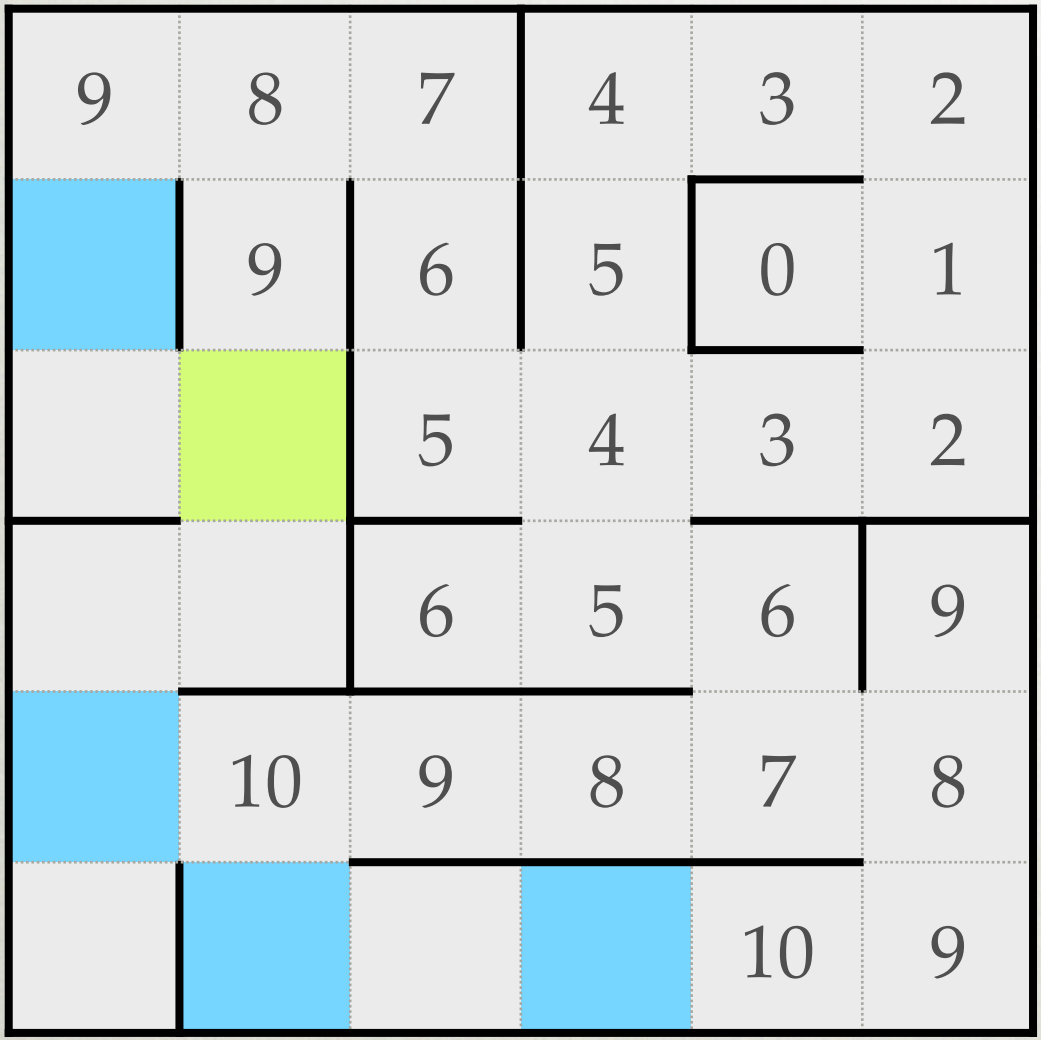


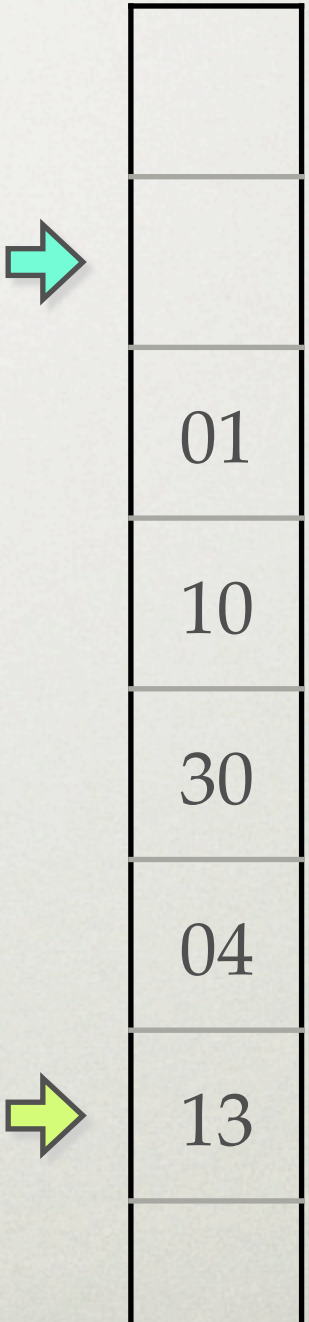


FULL STEP

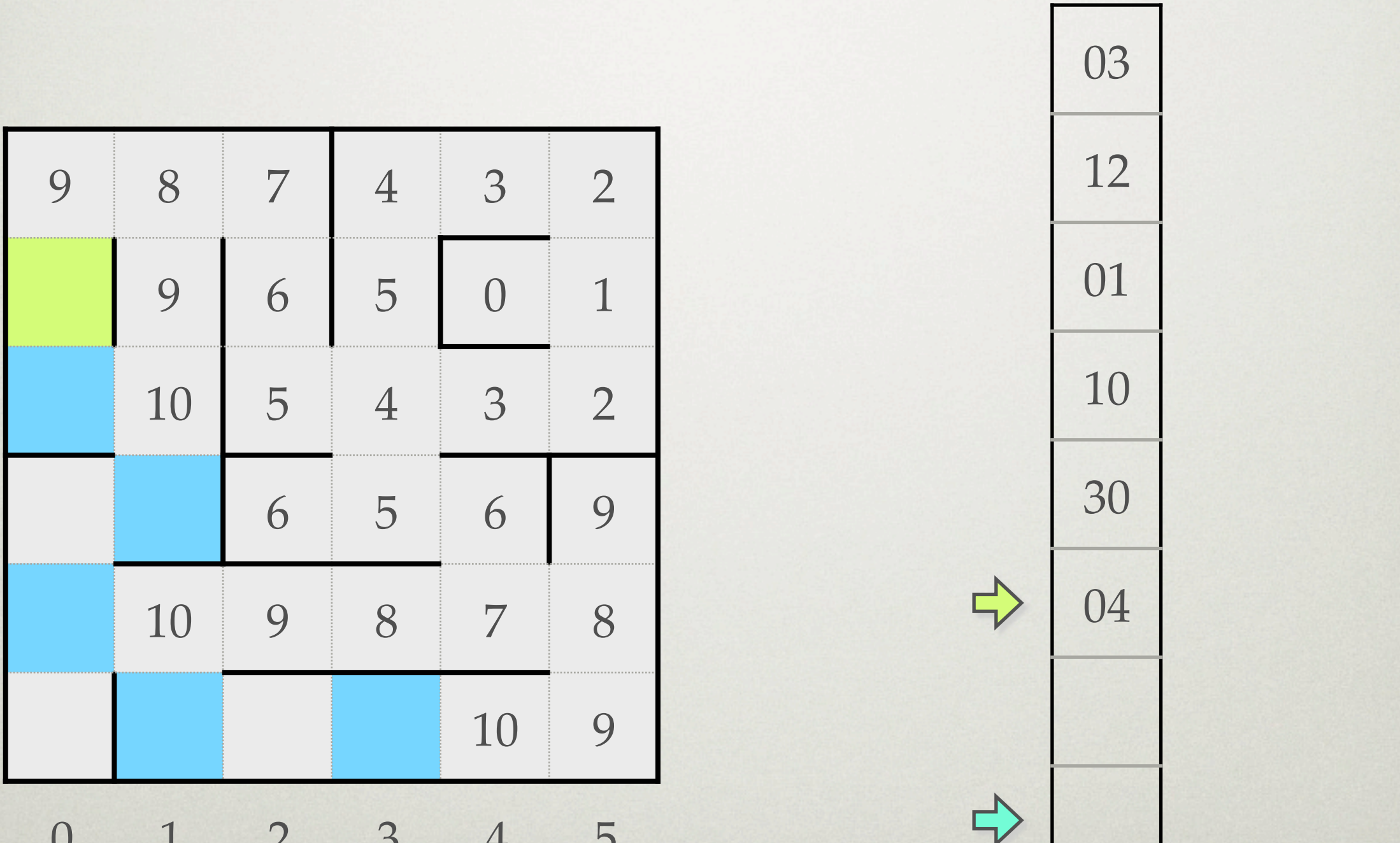


FULL STEP

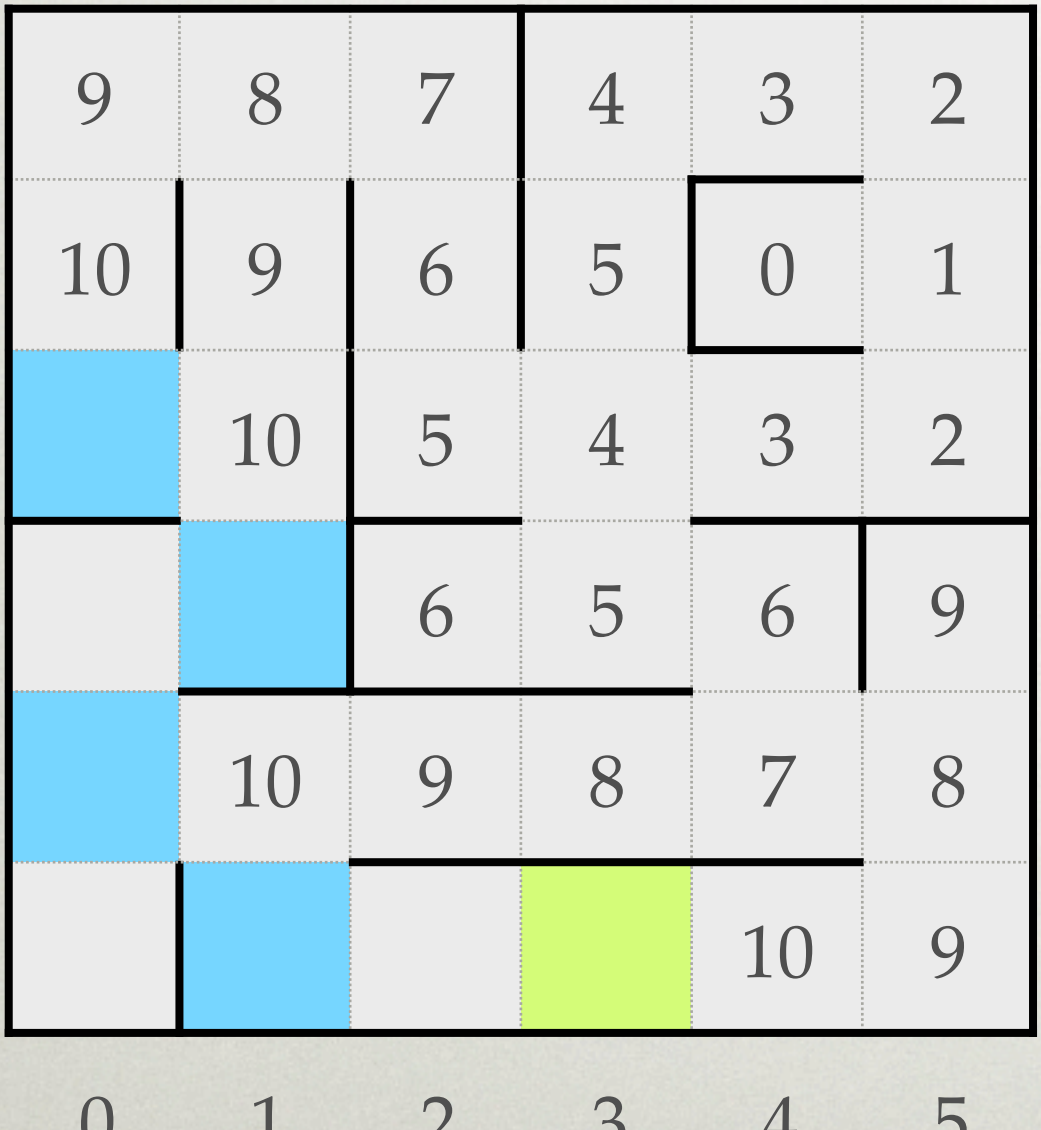


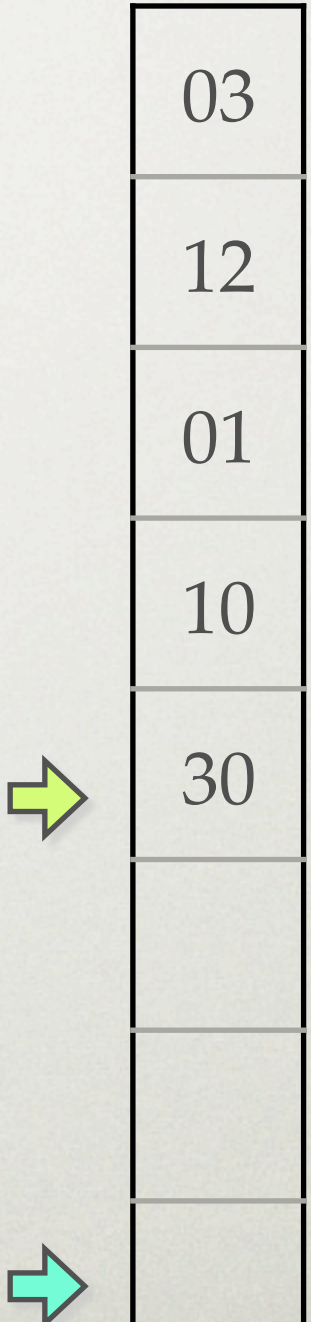


FULL STEP

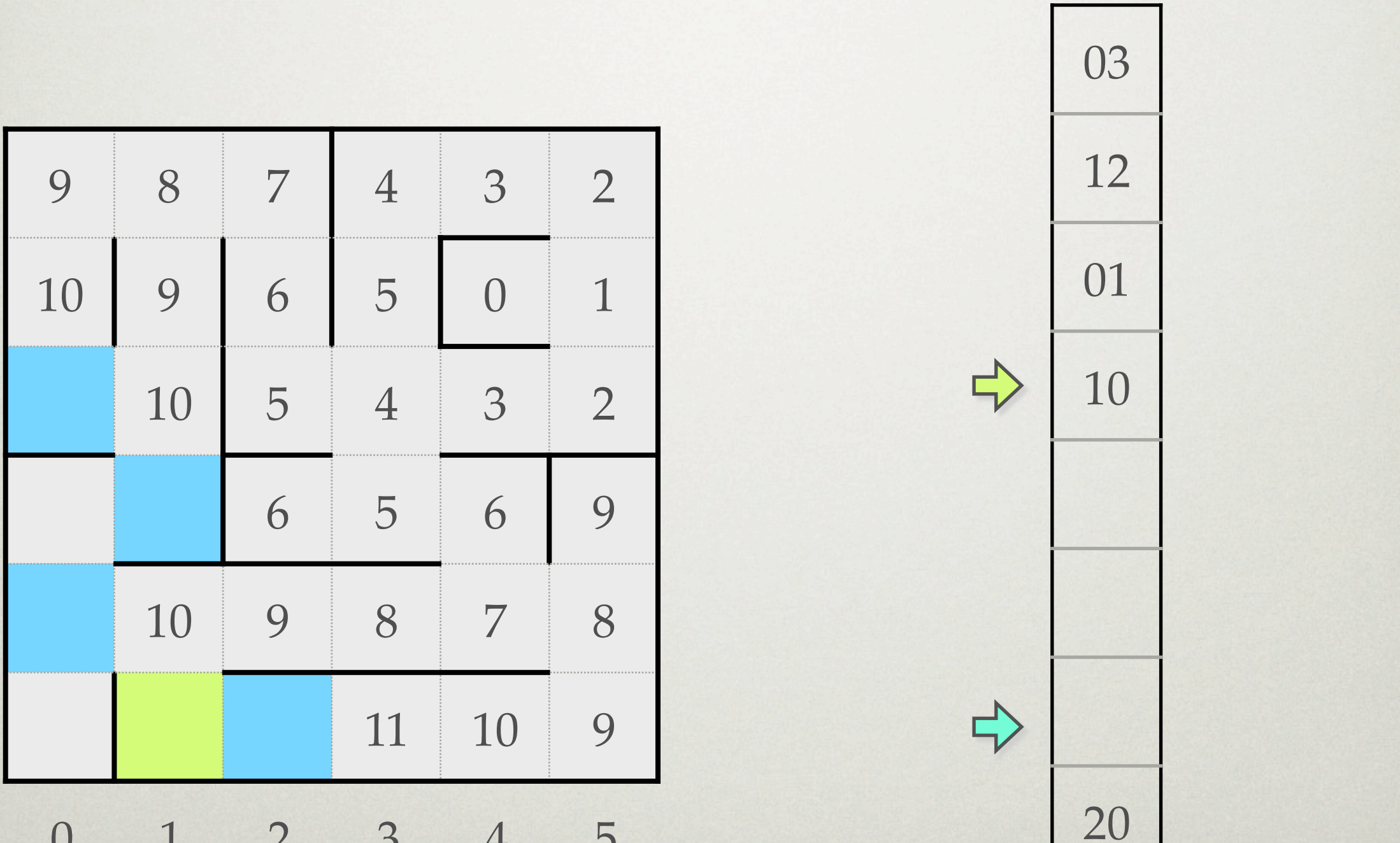


FULL STEP

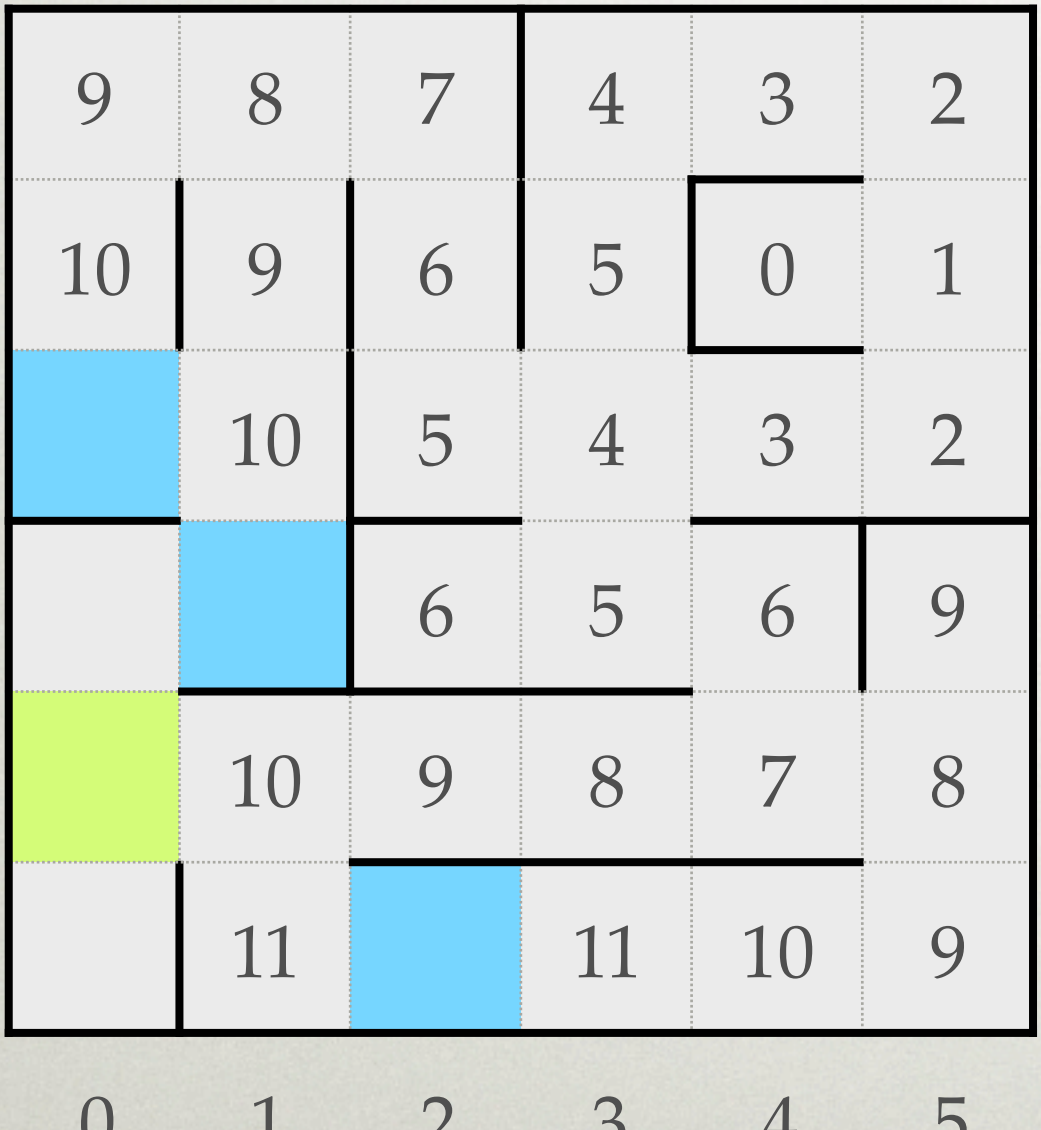




FULL STEP

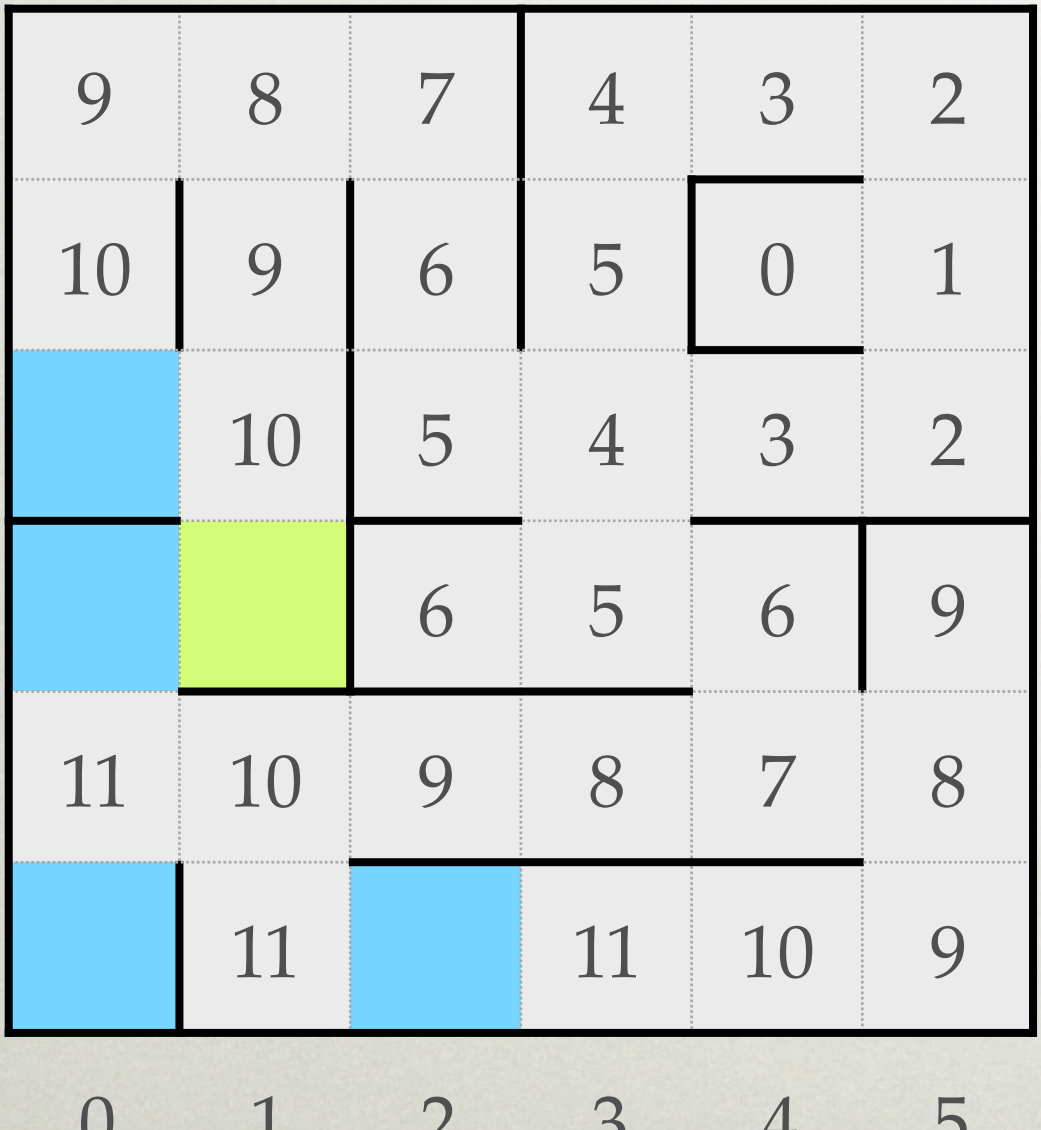

9 8 7 4 3 2
10 9 6 5 0 1
10 5 4 3 2
6 5 6 9
10 9 8 7 8
11 10 9
0 1 2 3 4 5
03
12
01
10
20

FULL STEP


9 8 7 4 3 2
10 9 6 5 0 1
10 5 4 3 2
6 5 6 9
10 9 8 7 8
11 11 10 9
0 1 2 3 4 5

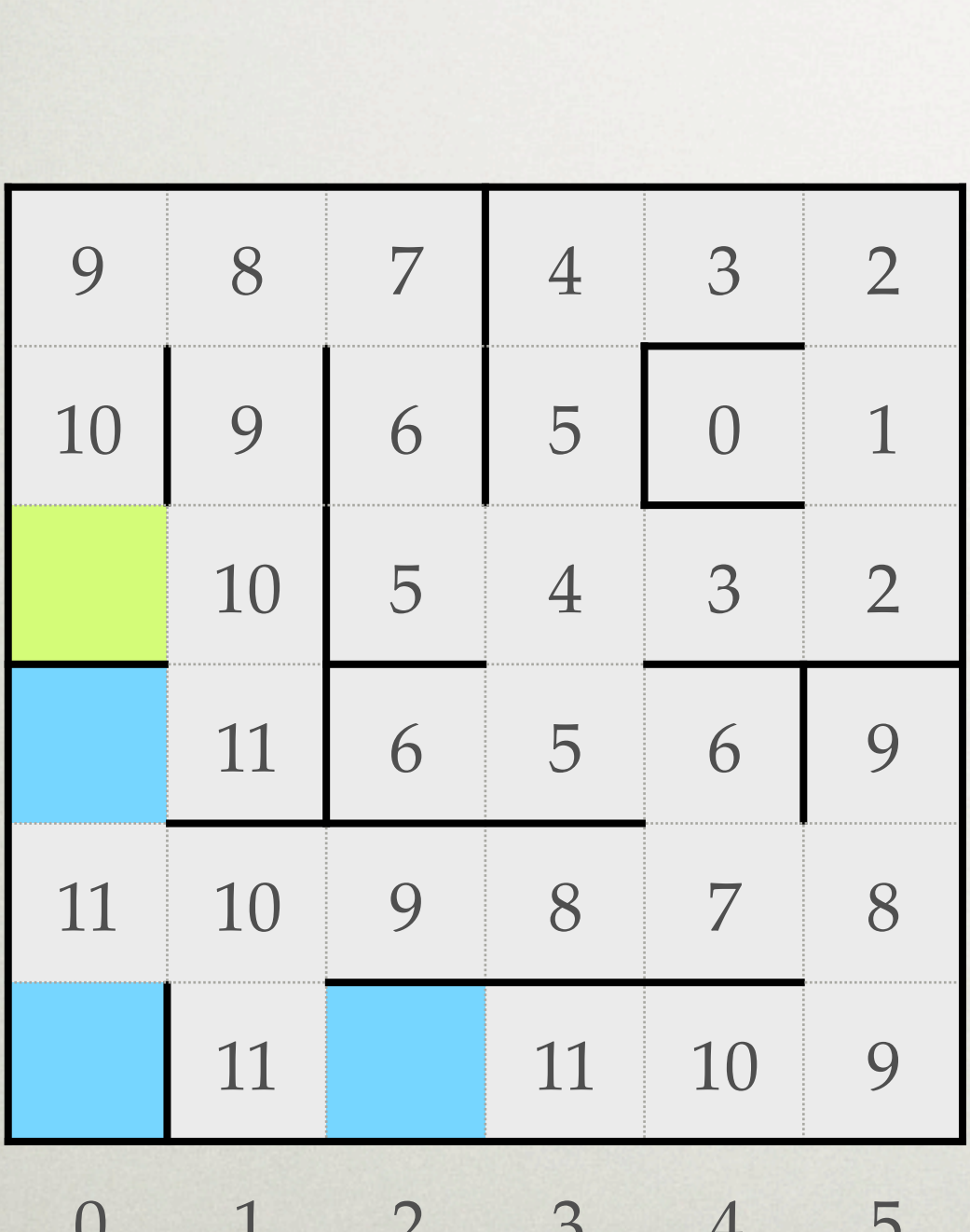

03
12
01
20

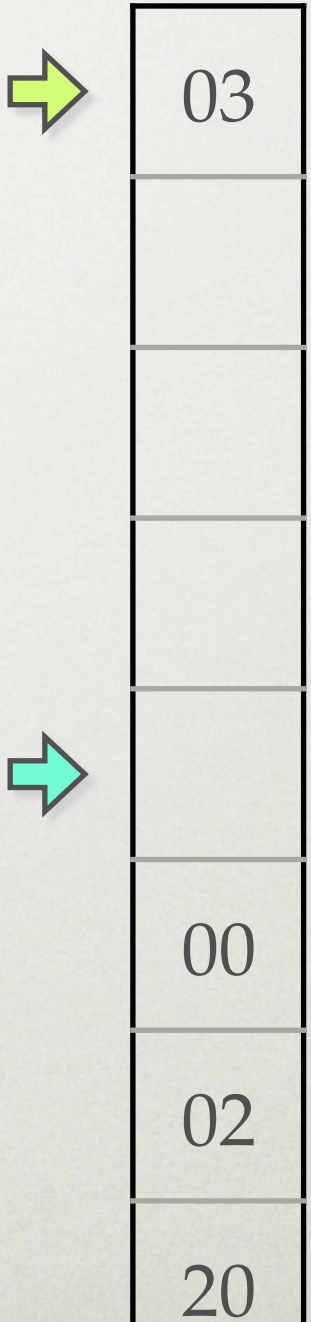
FULL STEP


9 8 7 4 3 2
10 9 6 5 0 1
10 5 4 3 2
6 5 6 9
11 10 9 8 7 8
11 11 10 9
0 1 2 3 4 5

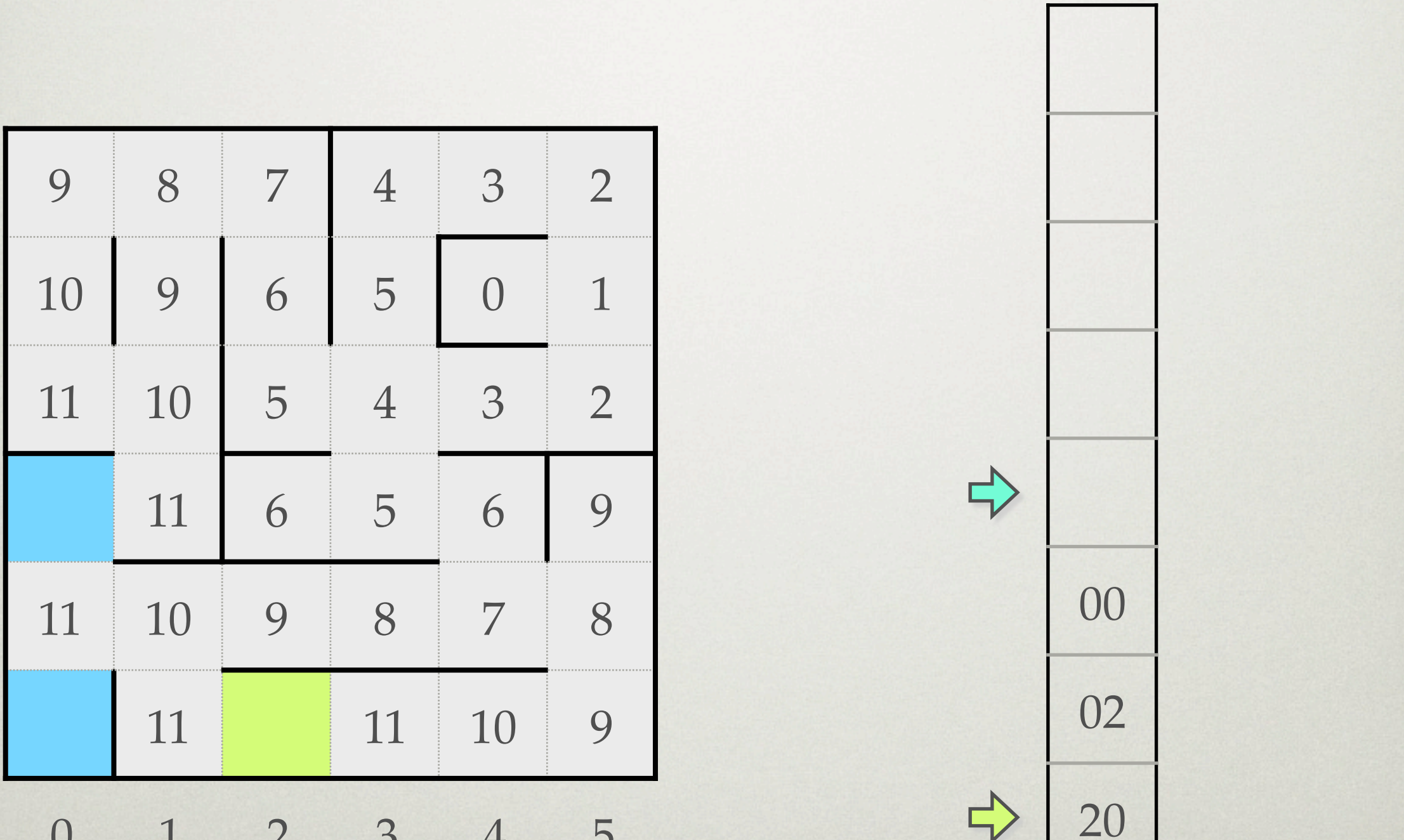

03
12
00
02
20

FULL STEP

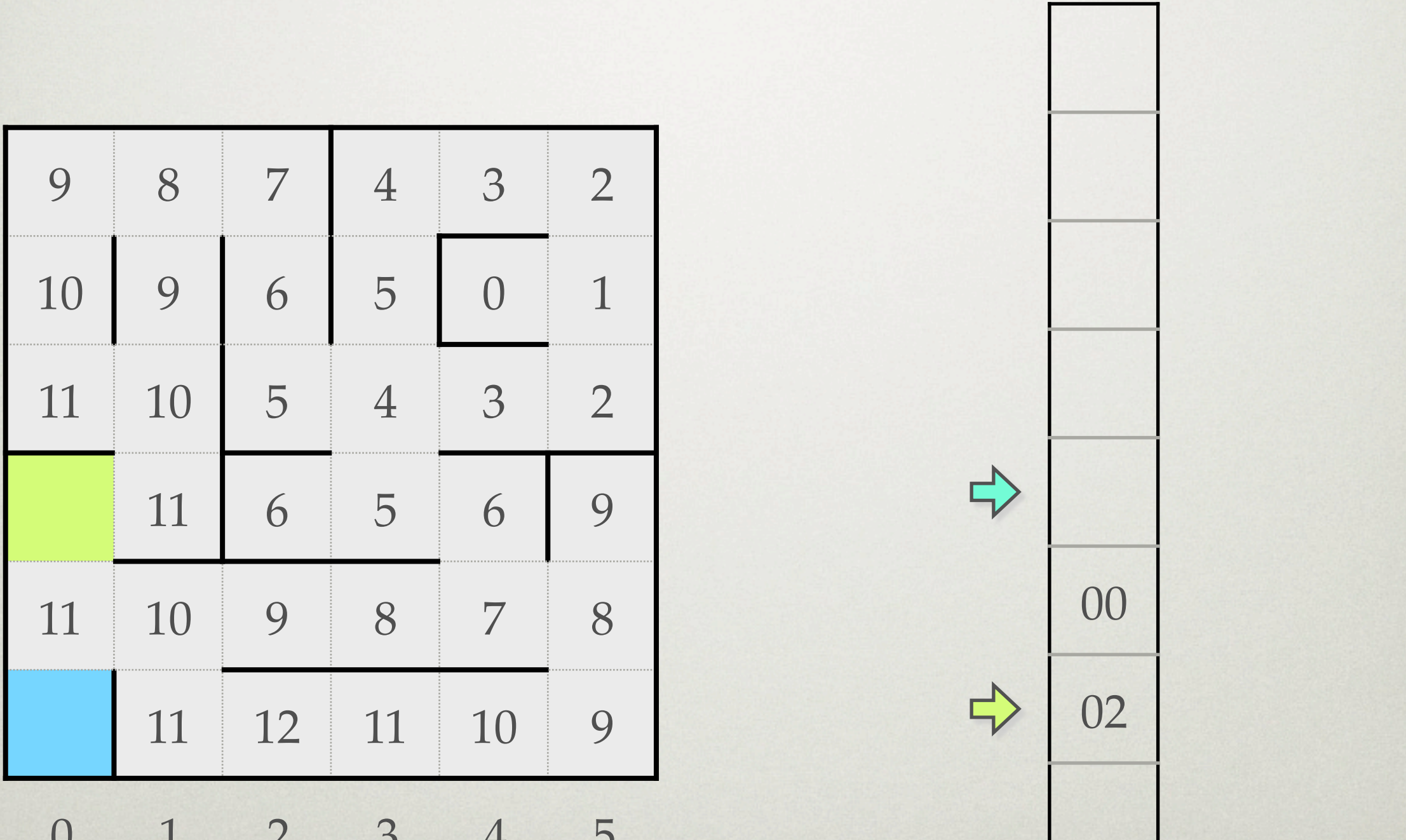

9 8 7 4 3 2
10 9 6 5 0 1
10 5 4 3 2
11 6 5 6 9
11 10 9 8 7 8
11 11 10 9
0 1 2 3 4 5


03
00
02
20

FULL STEP


9 8 7 4 3 2
10 9 6 5 0 1
11 10 5 4 3 2
11 6 5 6 9
11 10 9 8 7 8
11 11 11 10 9
0 1 2 3 4 5
00
02
20

FULL STEP

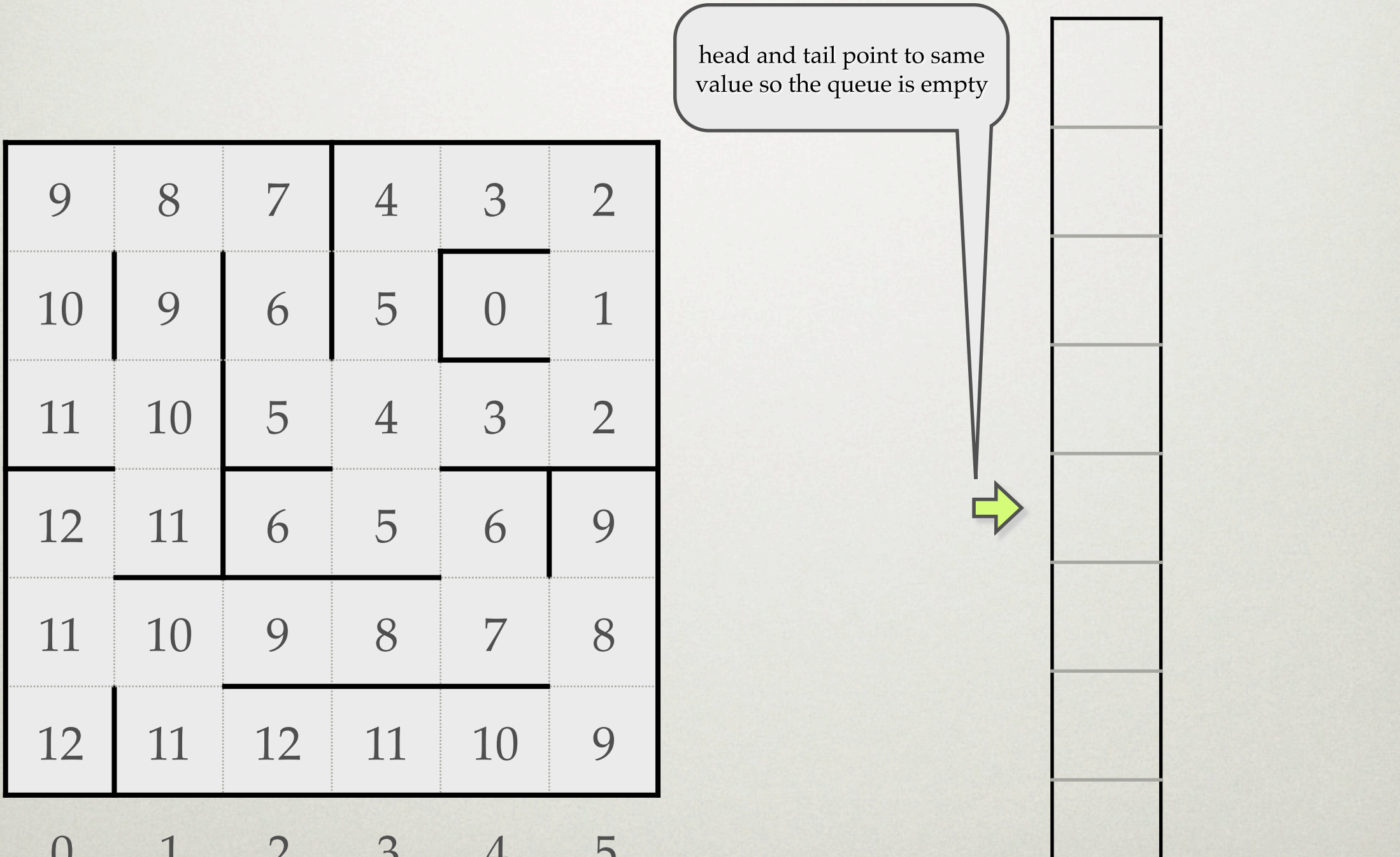

9 8 7 4 3 2
10 9 6 5 0 1
11 10 5 4 3 2
11 6 5 6 9
11 10 9 8 7 8
11 12 11 10 9
0 1 2 3 4 5
00
02

FULL STEP


9 8 7 4 3 2
10 9 6 5 0 1
11 10 5 4 3 2
12 11 6 5 6 9
11 10 9 8 7 8
11 12 11 10 9
0 1 2 3 4 5

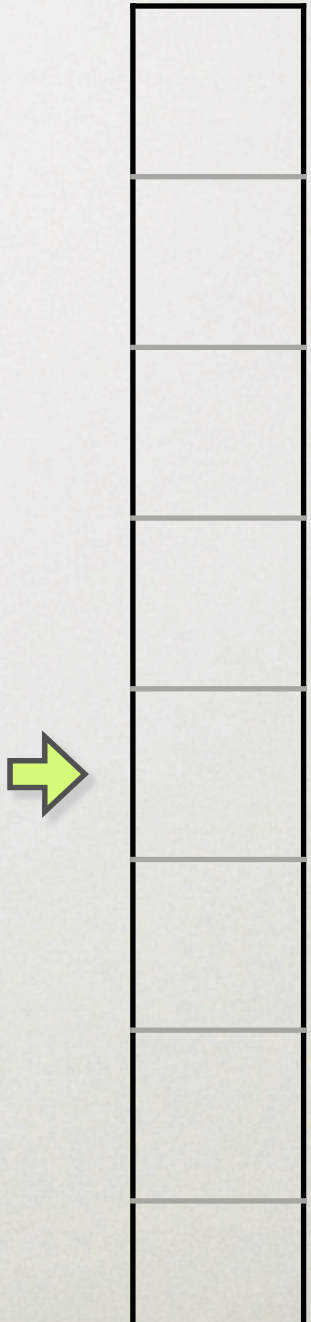

00

SOLVED


head and tail point to same value so the queue is empty
9 8 7 4 3 2
10 9 6 5 0 1
11 10 5 4 3 2
12 11 6 5 6 9
11 10 9 8 7 8
12 11 12 11 10 9
0 1 2 3 4 5

ROUTE TRACED

9	8	7	4	3	2
10	9	6	5	0	1
11	10	5	4	3	2
12	11	6	5	6	9
11	10	9	8	7	8
12	11	12	11	10	9



How fast is that?

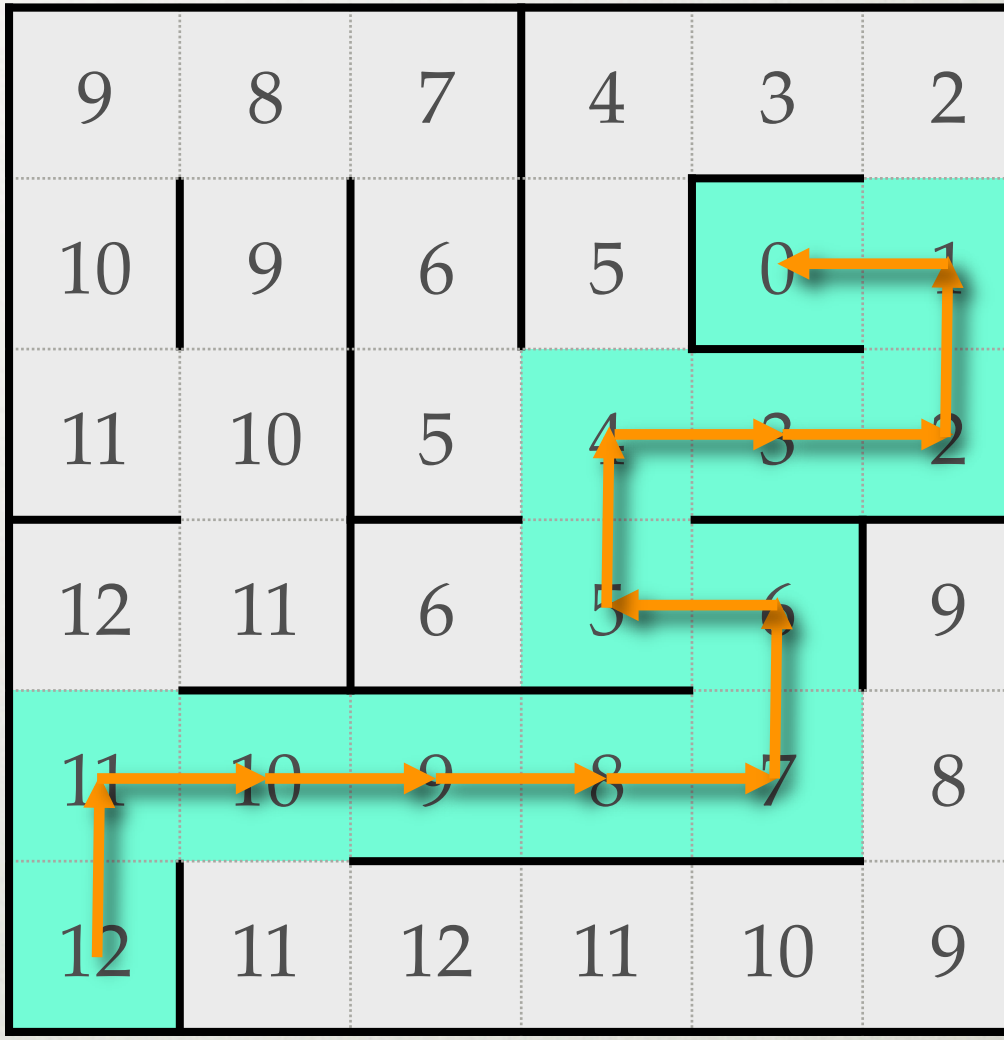
- dsPIC30F4011 @ 32MHz, C30
- Japan 2007 maze
- No optimisation
  - Simple flood: 41.91ms
  - Fast flood: 3.75ms, 139 words
- Optimisation O1
  - Simple flood: 24.19ms
  - Fast flood: 1.74ms, 63 words

How fast is that?

- PIC18F4525 @ 32MHz, C18
- Japan 2007 maze
- No optimisation
  - Simple flood: 46.9ms, 291 words
  - Fast flood: 5.95ms, 374 words
- Optimisation - full
  - Simple flood: 52.8ms, 166 words
  - Fast flood: 7.6ms, 196 words

CREATING THE PATH

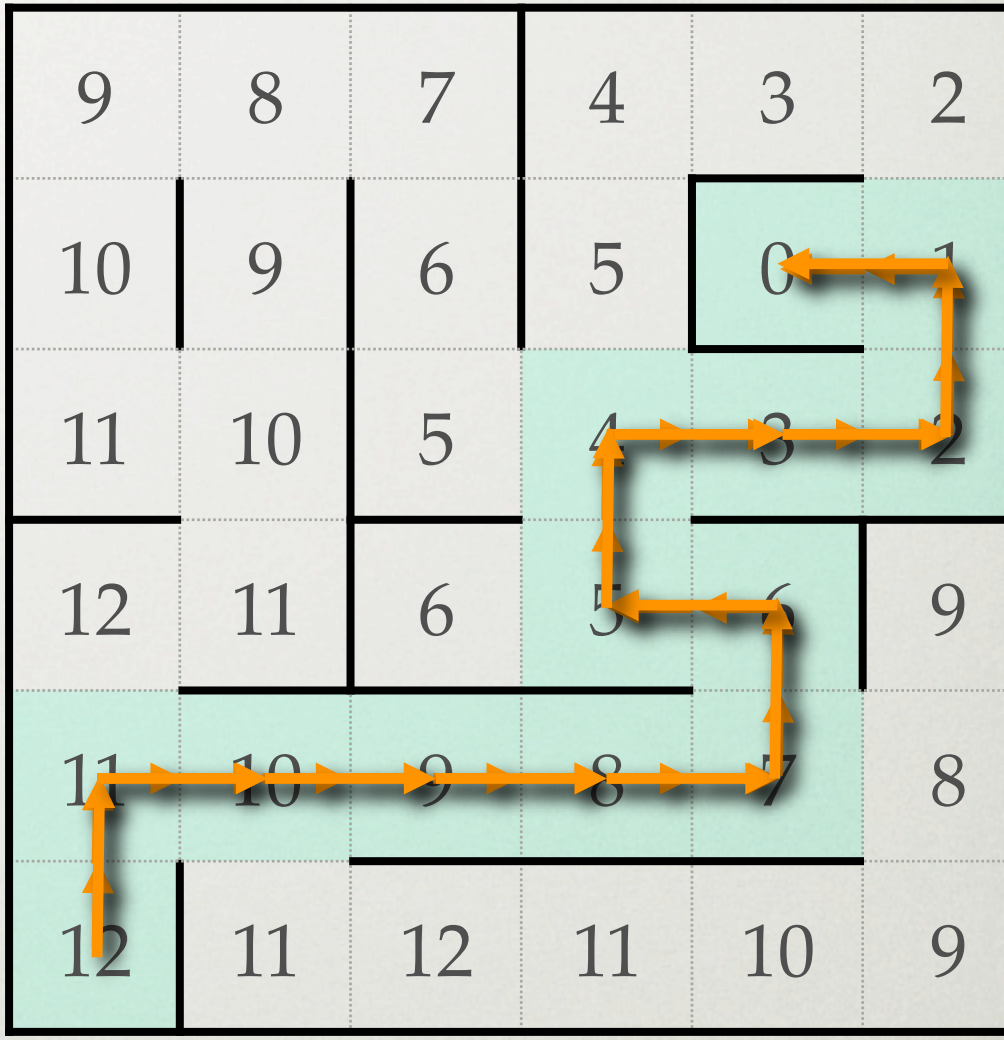
- Walk downhill
- Run length encode result



F RFFFF LF LF RF RFF LF LF
1R4L1L1R1R2L1L1

CREATING DIAGONALS

- Still downhill
- Pattern matching
- 19 patterns/rules
- 7 cell look-ahead
- messy at best
- Make more - do less
- Use half-cells?

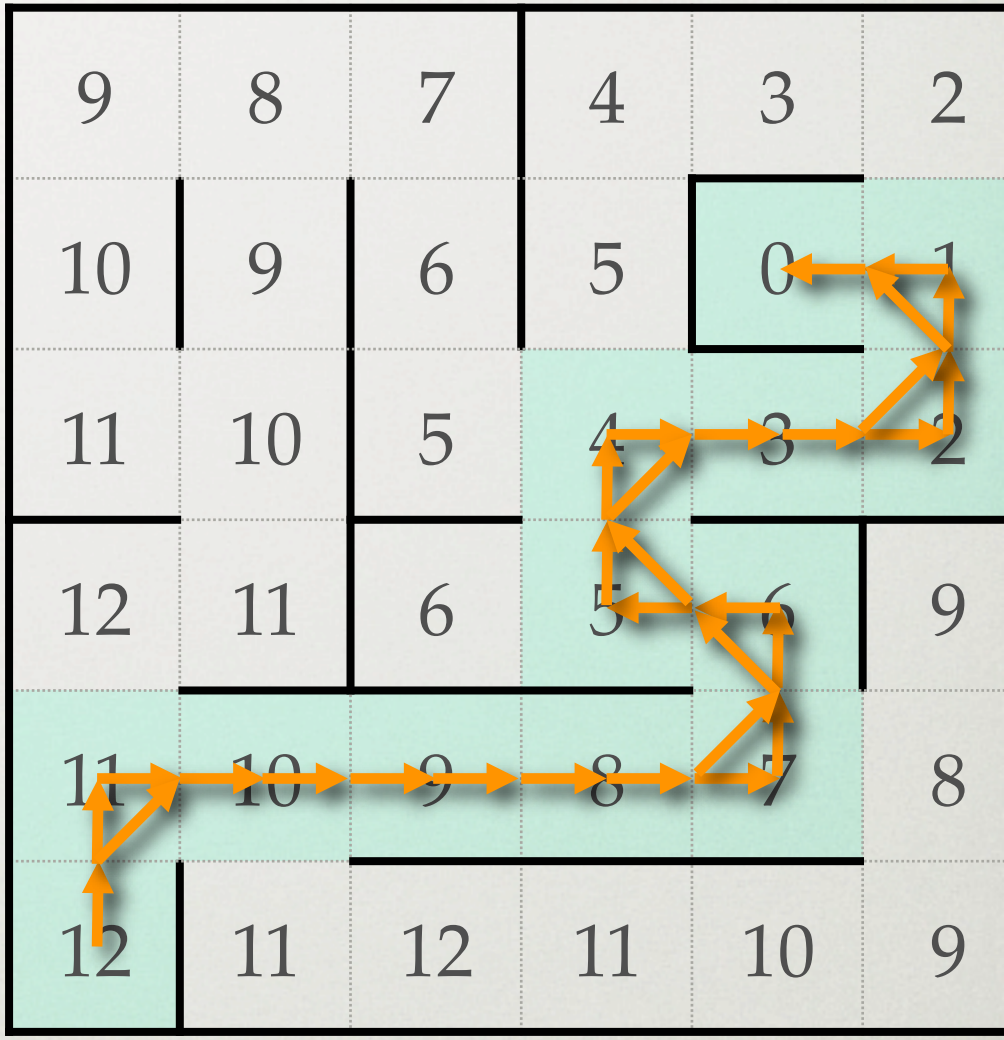


HHRHHHHHHHHLHHLHHRHHRHHHHHLHHLHH

1R4L1L1R1R2L1L1

CREATING DIAGONALS

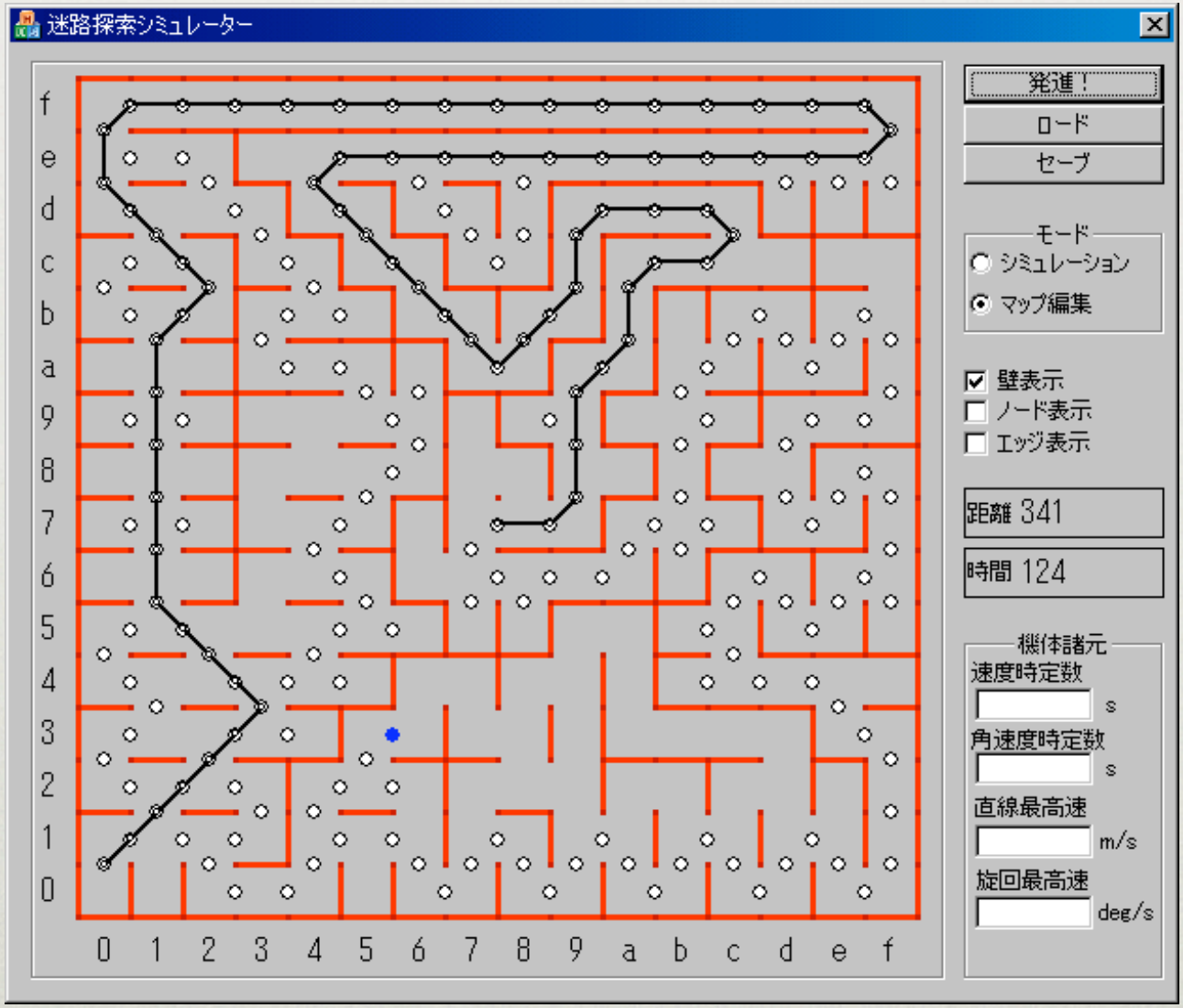
- Still pattern matching
- 7 patterns
- 6 cell look-ahead
- Turns are left as an exercise for the reader (this is not a good example)

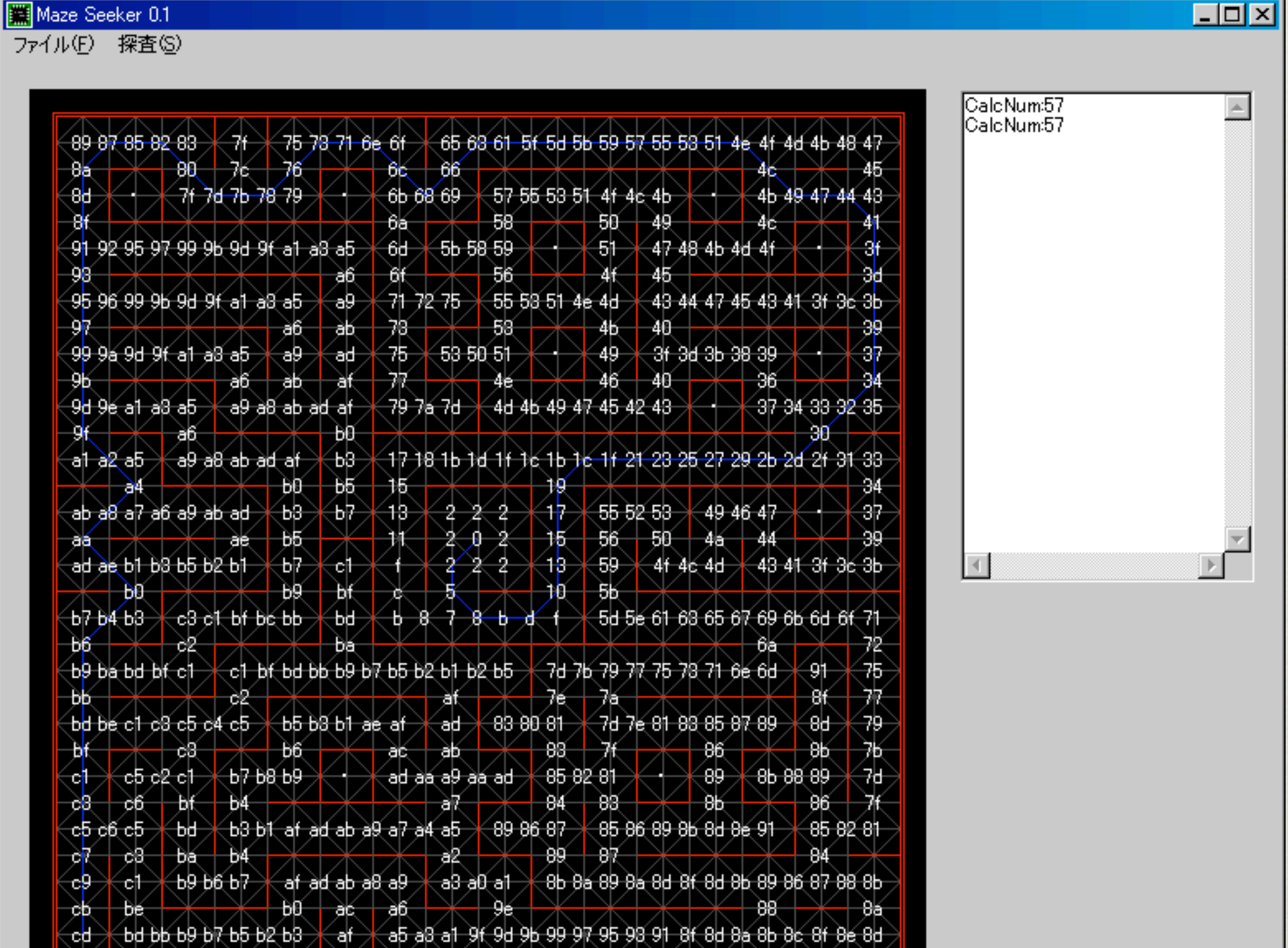


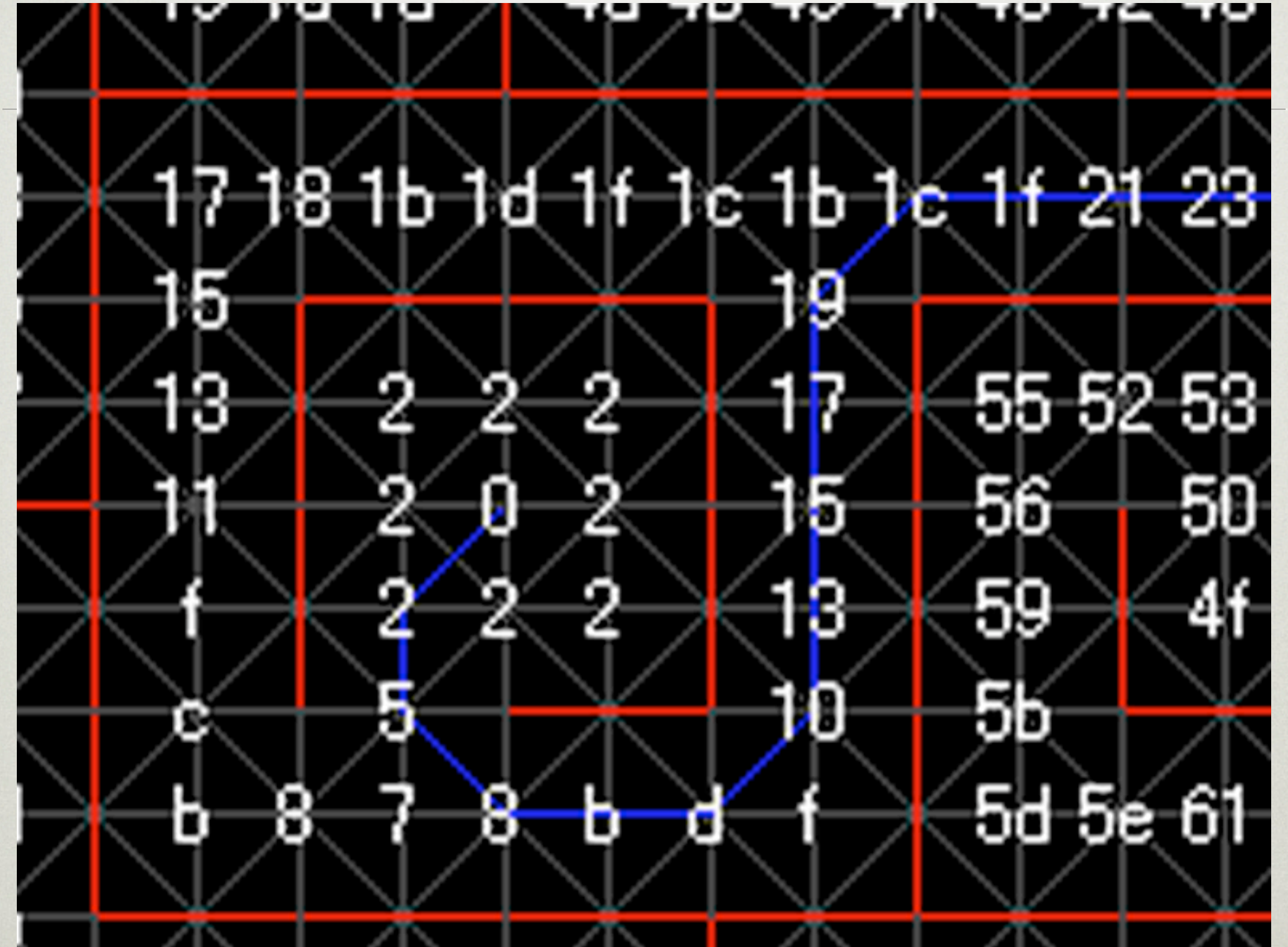
H HRH HHHHHH HLH HLH HRH HRH HH HLH HLH H
H TDT HHHHHH TDT TDDT TDT HH TDT TDT H

HALF-CELL GRIDS

- It may be easier to work on a half-cell grid in the first place



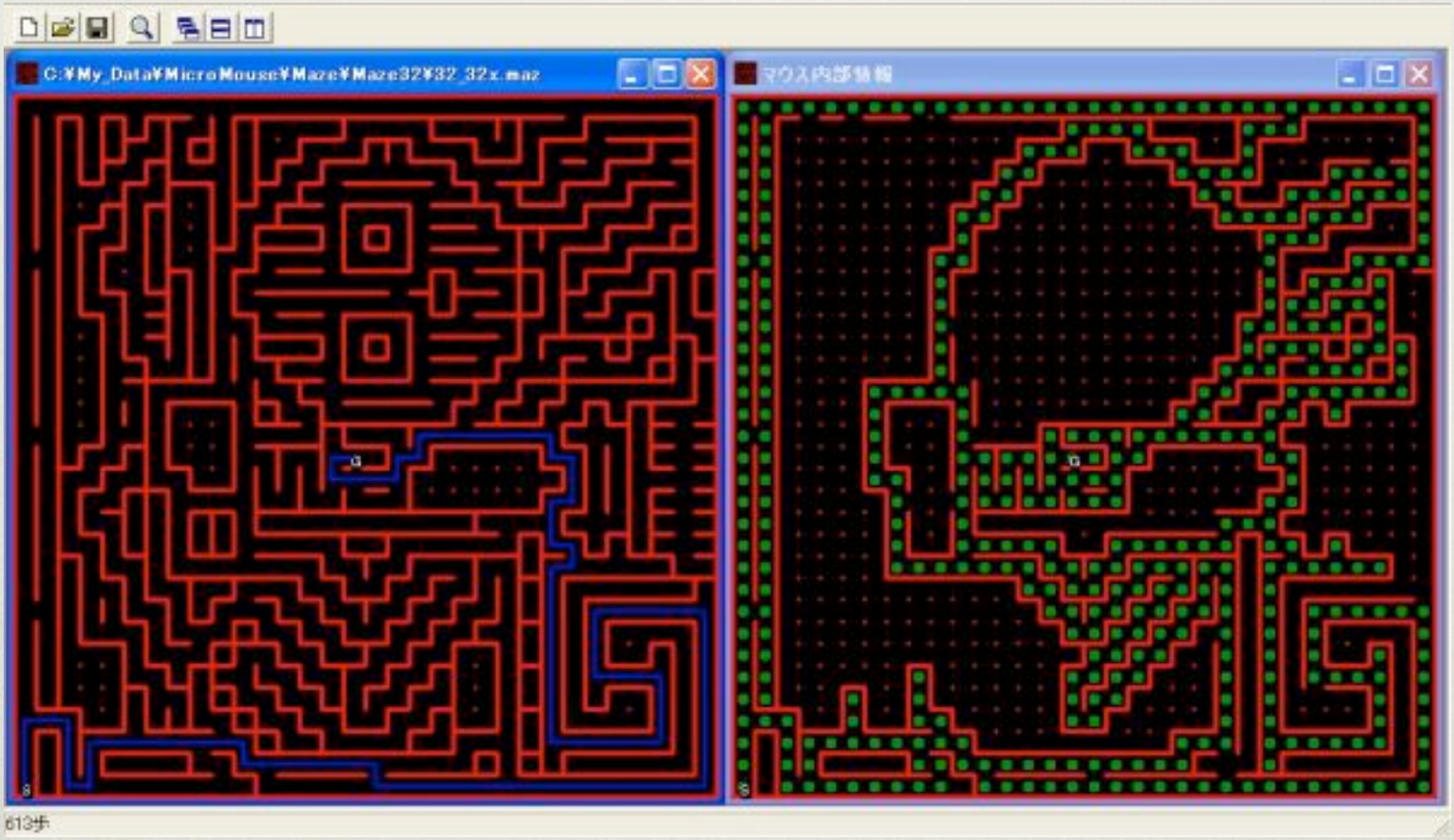

Maze Seeker 0.1
ファイル(F)　探査(S)
CalcNum:57
CalcNum:57


17 18 1b 1d 1f 1c 1b 1c 1f 21 23
15
19
13 2 2 2 17 55 52 53
11 2 0 2 15 56 50
f 2 2 2 13 59 4f
c 5 10 5b
b 8 7 8 b d f 5d 5e 61

FOUR TIMES AS BIG

http://www24.atwiki.jp/mm3sakusya/pages/28.html

FOUR TIMES AS BIG



http://www24.atwiki.jp/mm3sakusya/pages/28.html

Thank you

Peter Harrison

www.micromouseonline.com