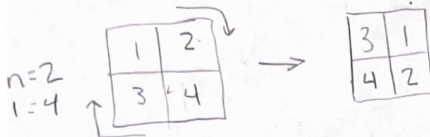


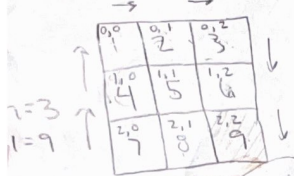
Rotate Tables Coding Challenge

From prompt:

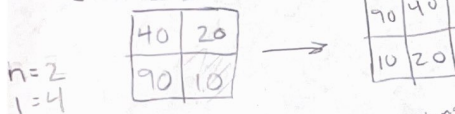


[1, 2, 3, 4]
[3, 1, 4, 2]

$n=3$ length=9



[1, 2, 3, 4, 5, 6, 7, 8, 9]
[4, 1, 2, 7, 5, 3, 8, 9, 6]



[40, 20, 90, 10]
[90, 40, 10, 20]

Hashmap
40 20 90 10
0 1 2 3
1 3 0 2

40 20 90 10
1 3 0 2

$i++$
 $z++$

Thoughts

- Infer rows and columns from input data
- maybe use hashmap to store indexes of each number
- $n = \text{number cols, rows} = \sqrt{\text{length}}$
- $z = 0$ if length is not square, false, if length is 1, return same
- func rotate(n, z) {
- for (i=0; i < n):
- for (j=0; j < n):
- if (i > n-z) and (i < j) shift down z spaces in list
- if (i < n-z) and (i > j) shift up z spaces in the list
- if (j < n-1-i) and (j > i-1) shift right (z+1)
- if (j > n-1-i) and (j < i+1) shift left (z-1)

$a_0 = [1, 2, 3, 4, 5, 6, 7, 8, 9]$
 $a_1 = [4, 1, 2, 7, 5, 3, 8, 9, 6]$

new example

$n=4$
 $l=16$



first col, last col, first row, mid section, last row

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
[5, 1, 2, 3, 9, 10, 6, 7, 11, 12, 13, 14, 15, 16, 4]

always shift left (except for first element (up))

