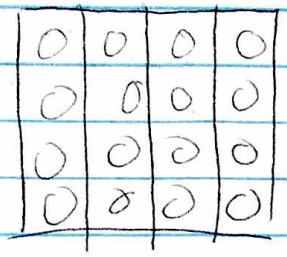
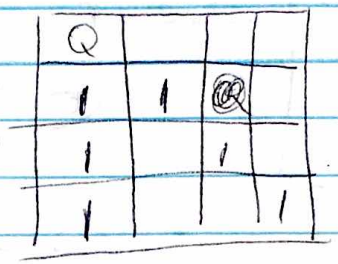


→ search 0 → for each queen column position in first row

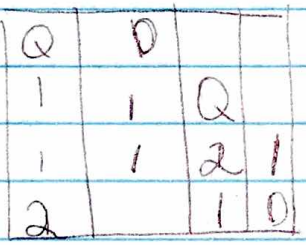
q | 0 | | | |

row = 0
→ place queen (0,0) →



→ make recursive call search(1)
t[1][0] == 1
t[3][1] == 1
t[1][2] == 0

→



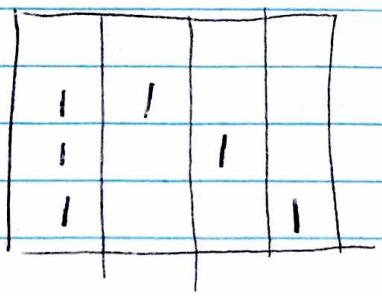
q | 0 | 2 | |

changeThreats(1, 2, 1)

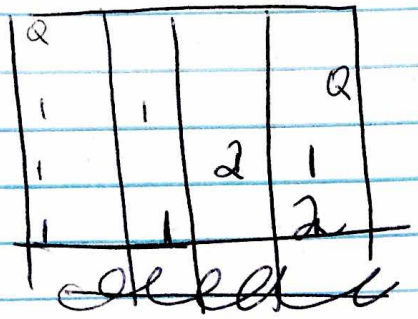
- search(2) no recursive calls made
Dead end; return from w/in search(1) call
- start undoing what we had done
- remove threats from placing Q in (1,2)

Backtrack!

→ Q



w/in the search (1) call
- place queen in (1,3) q [0|3| |]



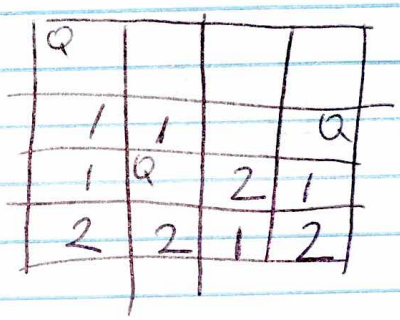
and update threats

make search(2) call again

- find place for queen [0|3|1|]

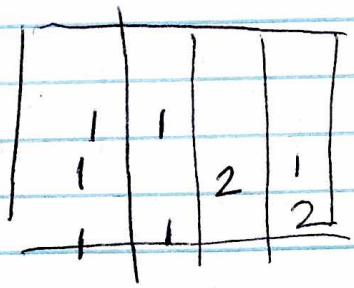
Update threats

search(3)



no recursive calls - dead end from w/in search(2)

back track & remove threats

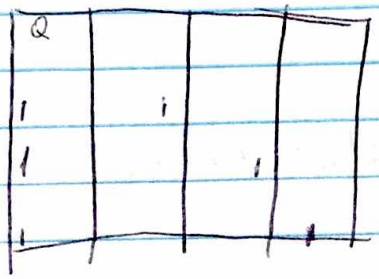


w/in search(2) call
no more places for queen in row 2 either q [0|3|4|]

so search(2) will return

w/in search(1) call

backtrack and remove threats



increment column and exhaust all positions in row 1

q [0|4|4|4]

· search(1) will return

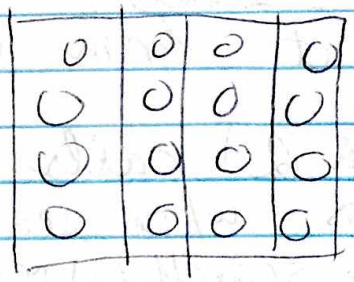
w/in search(0) → backtrack

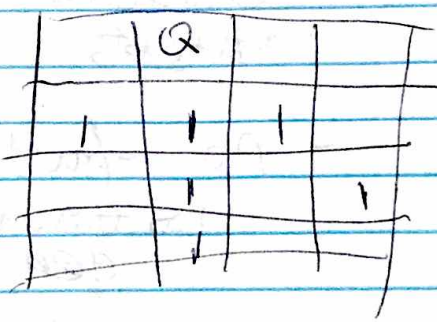
~~board of~~ threats of all 0 again

and start over placing Q = (0,1)

first column
0 row

q = [1|4|4|4]





search(1) call →

threats 4

search(2) call

q [1|3|1]

update

	Q		
1	1	1	Q
Q	1	1	2
	2		1

q [1|3|0]

↓ update threats

	Q		
1	1	1	Q
Q	1	1	2
1	3	Q	1

call search(3)

only one spot q [1|3|0|2]

No threats to update

search (4) print solution

no threats undo

return search (3)

- undo threats
- no further search(3) calls

= search(2) returns, ~~undoes threats~~ w/in search 1 undo threats from Q(1,3)

- no further search(2) calls

↳ ~~undoes threats~~ w/in search(1)

~~q[1] = 4~~ q[1] = 4 ⇒ search(1) returns

⇒ undoes threats from Q = (0,1) → start w/ Q = (0,2)