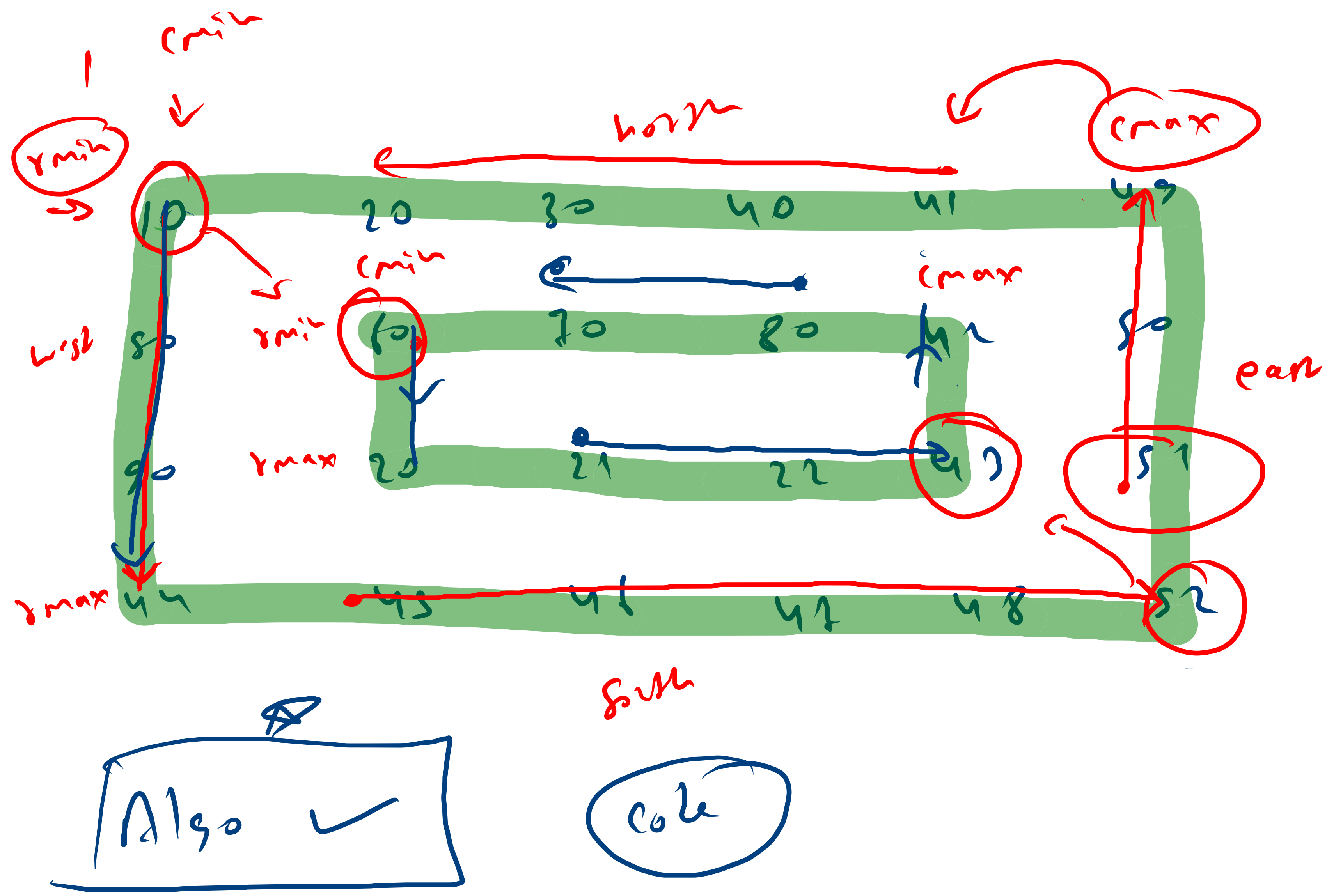


rmin cmin done = 0
rmax cmax total = n*m

while (d < t) {

west
for r <- rmin to rmax
[r][cmin]; d++

south
for c <- cmin+1 to cmax
[rmax][c]; d++

if(d==t) break;

east
for r <- rmax-1 to rmin
[r][cmax]; d++

north
for c <- cmax-1 to cmin+1
[rmin][c]; d++

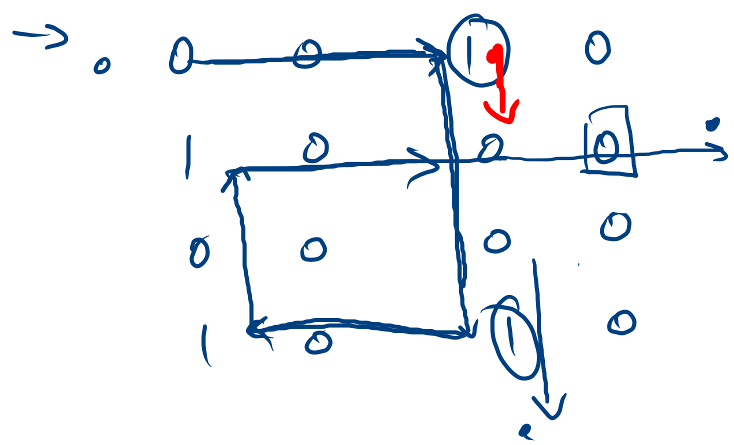
=> rmin++ rmax--
cmin++ cmax--

}

i j Q asked

logic ✓

Implement



1 ☆

go c/w clockwise

last valid index

1 → row
3 → col

dir
0 east
1 south
2 west
3 north
0 east

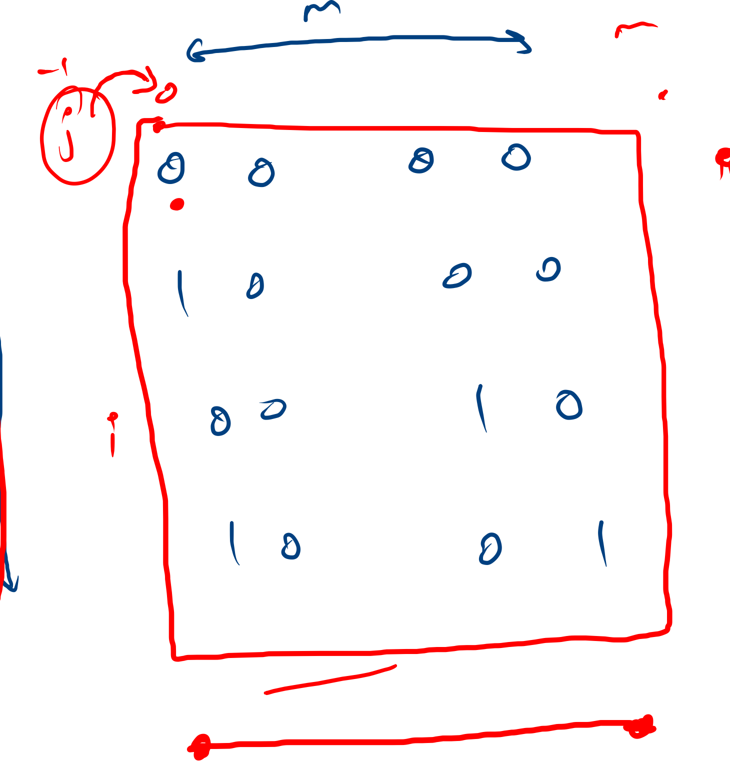
j++	j==m
i++	i==n
j--	j==-1
i--	i==-1
j++	

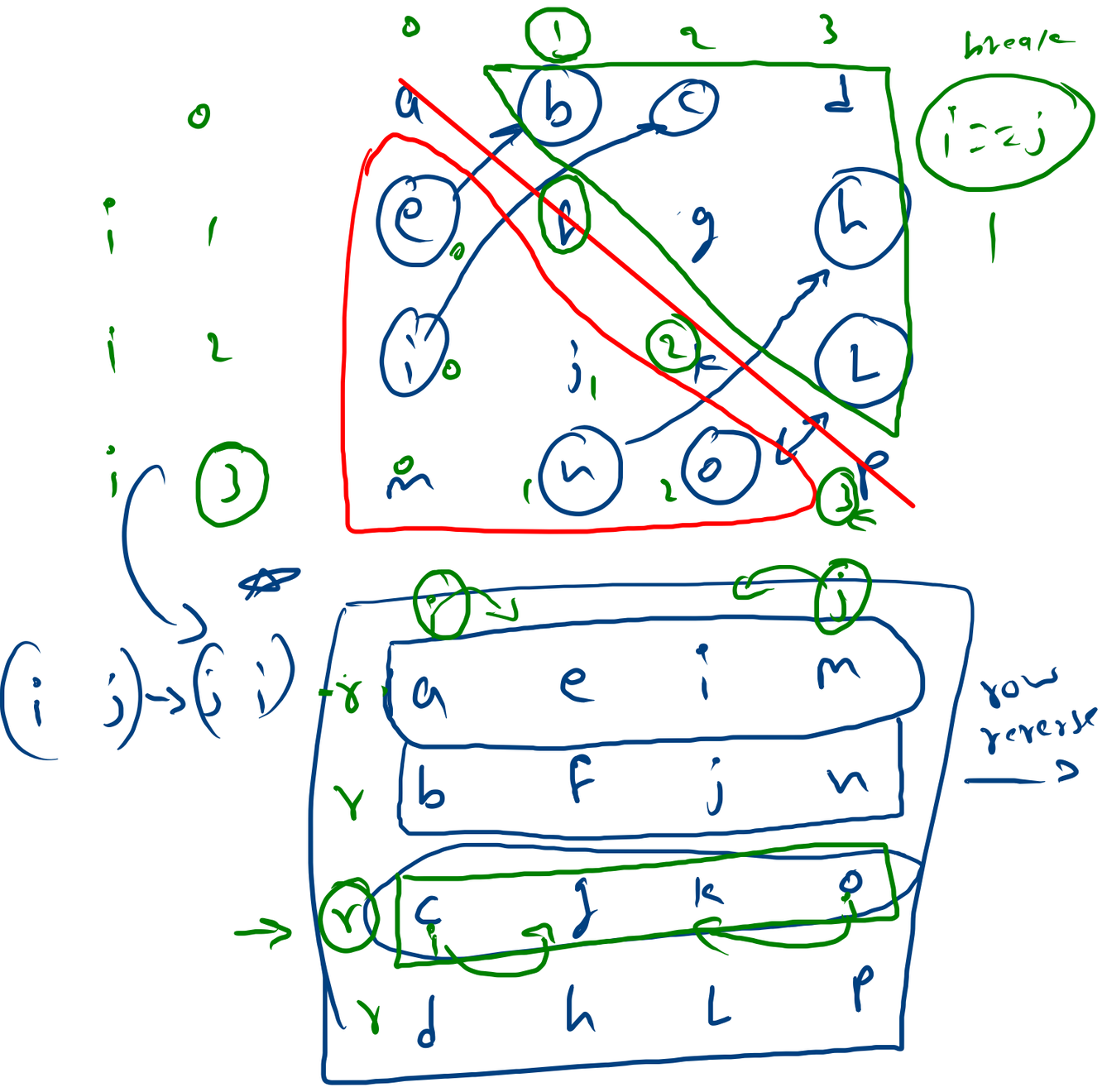
```
while(1==1) {   true
    move
    if( invalid) {
        correct
        break;
    }
}
```

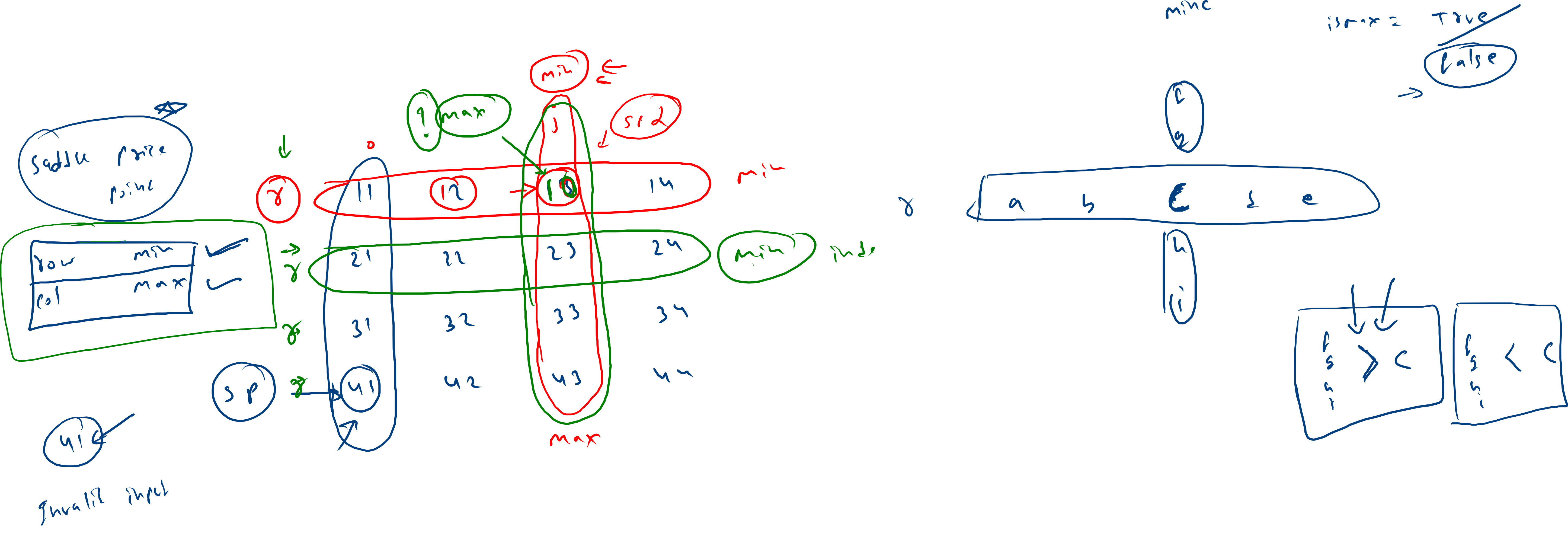
priority

move next

direction




break
i==j
row reverse
clockwise



saddle point
row min
col max
sp
41
invalid input
min
max
11 12 13 14
21 22 23 24
31 32 33 34
41 42 43 44
a b c d e
isMax = True
False

multi