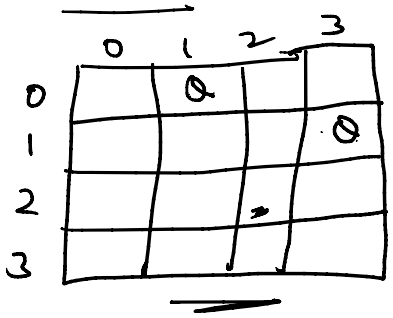
23-Jan-2022

23 January 2022 20:07

N-Queens :- https://leetcode.com/problems/n-queens/

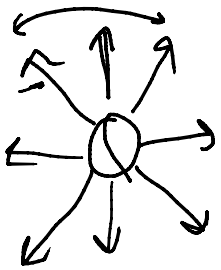
n = 4

n x n



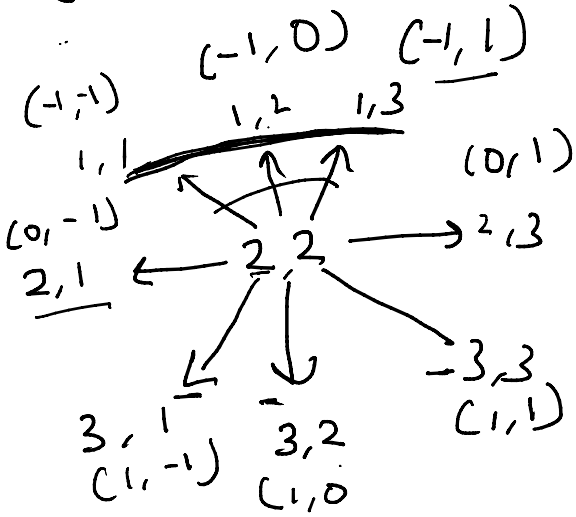

2,2

n x n



n Queens

dir = 8 steps




static int[][] dir = { {-1,-1}, {-1,0}, {-1,1} };

isValid(chess, r, c) {

for (i:0 to dir.length-1) {

tr = r | mr = dir[i][0]

tc = c | mc = dir[i][1]

while (tr+mr ≥ 0 && tc+mc ≥ 0 &&

tc+mc < chess.length) {

if (chess[tr+mr][tc+mc] == 'Q')

return false;

tr += mr; tc += mc

}

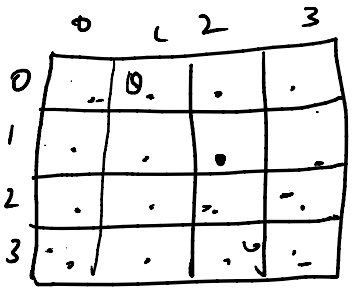
return true;

}

1, 'Q'

return true;

}



List<List<String>> nQueens(chess, r, c,

1, 0

canPlace = canBePlaced(chess, r, c);

if(canPlace) {

chess[r][c] = 'Q';

nQueens(chess, r+1, 0

chess[r][c] = '.';

}

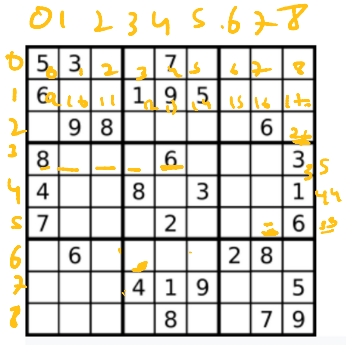
if(c+1 < chess.length) {

nQueens(chess, r, c+1);

}

}

9 x 9



1. Each row must contain the digits 1-9 without repetition.
2. Each column must contain the digits 1-9 without repetition.
3. Each of the nine 3 x 3 sub-boxes of the grid must contain the digits 1-9 without repetition.

1 - 9

3, 4

31th

31/9 = 3,

31%9 = 4

52nd

52/9 = 5,

52%9 = 7

(5, 7)

6 ^ 3

7, 3

sr = r - (r%3);

valid(board) {

for(i=0 to 81) {

```
sr = r - (r%3);
sc = c - (c%3);
```

```
for(i=0 to 81){
    int r = i/9;
        c = i%9;
    if(board[r][c] != '.'){
```

```
boolean solveSudoku(board, n)
if(n == 81) return true;
    r = n/9;
    c = n%9;
if(board[r][c] != '.'){
    solveSudoku(board, n+1);
```

0 1 2 3 4 5 6 7 8

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 5 | 3 | | | 7 | | | | |
| 1 | 6 | | | 1 | 9 | 5 | | | |
| 2 | | 9 | 8 | | | | | 6 | |
| 3 | 8 | | | | 6 | | | | 3 |
| 4 | 4 | | | 8 | | 3 | | | 1 |
| 5 | 7 | | | | 2 | | | | 6 |
| 6 | | 6 | | | | | 2 | 8 | |
| 7 | | | | 4 | 1 | 9 | | | 5 |
| 8 | | | | | 8 | | | 7 | 9 |

n == 81

```
    }
    for(1 to 9){
if(isValid(board, r, c, i)){
    board[r][c] = i + '0';
if(solveSudoku(board, n+1)){
    return true
}
    board[r][c] = '.';
}
}
return false;
}
```

HashMap : Key, Value

Put :- O(1)

get :-

remove :-

containsKey :-

size() :-

LL[] buckets;

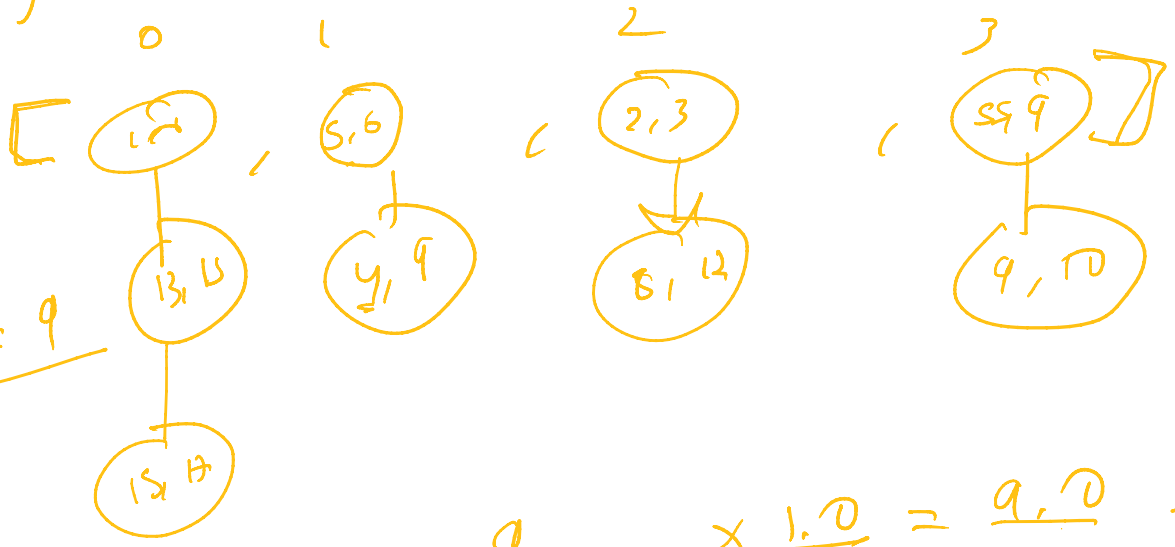
2

size() :-

Custom HM {

LL[] buckets ;
int size;

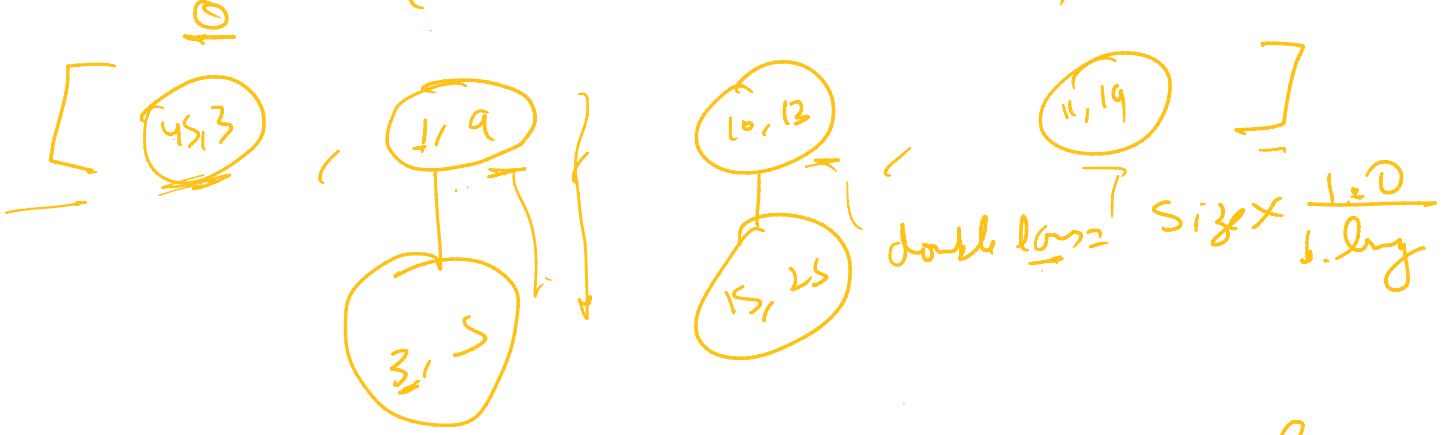
}




(1,2), (1,5), (3,5)

$\frac{9}{4} \times \frac{1.0}{4} = \frac{9.0}{4} = 2.25$

$2.25 > 2$




double long = size $\times \frac{1.0}{0.long}$

0 - (buckets.leng - 1)

0 - 3 = 1