

21

May

21st Wk • 141-224

2022

147

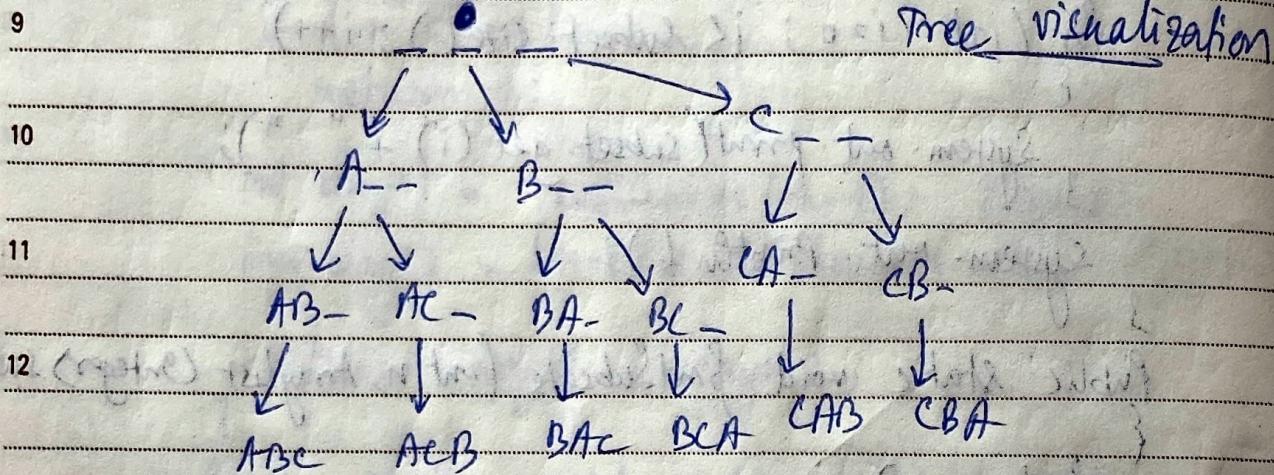
Saturday

Backtracking

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S			

✓ Find all possible solutions (and use the one we want)

(Q) E.g. → ABC → find all possible combinations



1 → class Backtracking {

2     Public static void printPermutation(String str, String perm, int idx)

3     {

4         if (str.length() == 0)

5         System.out.println(perm);

6         return;

7         for (int i=0; i&lt; str.length(); i++)

8             char currChar = str.charAt(i);

9             String newStr = str.substring(0, i) + str.substring(i+1);

10             printPermutation(newStr, perm + currChar, idx+1);

11 }

12     }

13     String str = "ABC";

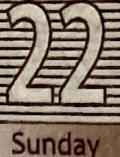
14     printPermutation(str, "", 0);

2022

JUNE
1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30
M T W T F S S M T W T F S S

May

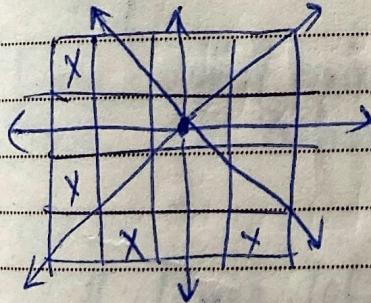
21st Wk • 142-223



148

if  $\rightarrow$  "ABC"opp  $\rightarrow$  ABC, ACB, BAC, BCA, CAB, CBA.Time complexity  $\rightarrow O(n \times n!)$ , where  $n = \text{no. of chars.}$ To reach 1 possible answer, we need to traverse  $n$  steps, and no. of possible combinations  $= n!$ .  $\therefore O(n \times n!)$ .12. 3) N - QueensN X N chessboardN Queens

Print all Solutions where queens are safe.

Note:- In a  $1 \times 1$ ,  $2 \times 2$ ,  $3 \times 3$  chessboard, only 1 queen can survive. So, we shall start with a  $4 \times 4$  chessboard

• queen can move in all 8 directions at all rooms.

So, boxes with 'X' are the safe options for the other queens.

Now, we will start by placing Queens in different columns.

		B3	
	B1		
			B4
	B2		

→ result 1,

23

May

22nd Wk • 143-222

Monday

S	T	M	T	W	T	F	S	E	T	1	2	3	4	5	6	7	8					
9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	T	W	T	F	S	S	M	T	W	T	F	S										



→ result ?

8  
9

Even after toying other combinations, we find that these

are only 2 possible solutions in a  $4 \times 4$  chessboard.

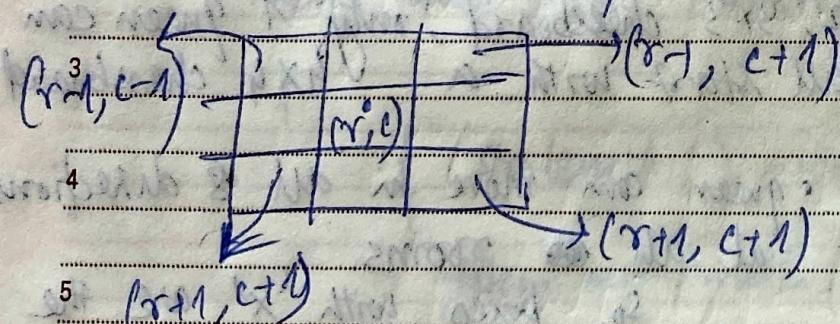
11 Input:- 4

12 Output:-  $\{["\cdot Q ..", ". . . Q", "Q . . .", "... Q ."],$  $["\cdot .. Q.", ". Q . .", "... Q .", ". . Q ."]\}$ 

✓ Answers are stored in a list of list of string.

✓ . remains represents empty spaces.

✓ answers are written from row perspective.



6 → class Solution {

public boolean isSafe (int row, int col, char[][] board)

// horizontal

for (int j = 0; j &lt; board.length; j++)

if (board[row][j] == 'Q')

return false;

}

2022

JUNE	1	2	3	4	5	6	7	8	9	10	11	12	
13	14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30										
M	T	W	T	F	S	S	M	T	W	T	F	S	

May

22nd Wk • 144-221

150

24

Tuesday

// vertical

```
for (int i = 0; i < board.length; i++) {
    if (board[i][col] == 'Q') {
        return false;
    }
}
```

// upper left

int r = row;

```
for (int c = col; c >= 0 && r >= 0; c--, r--) {
    if (board[r][c] == 'Q') {
        return false;
    }
}
```

// upper right

r = row;

```
for (int c = col; c < board.length && r >= 0; r++, c++) {
    if (board[r][c] == 'Q') {
        return false;
    }
}
```

// lower left

r = row;

```
for (int c = col; c >= 0 && board.length > r; r++, c--) {
    if (board[r][c] == 'Q') {
        return false;
    }
}
```

25

May

22nd Wk • 145-220

Wednesday

2022 151  
MAY

S	F	S	B	S	1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21
23	24	25	26	27	28	29	30	31				
M	T	W	T	F	S	S	M	T	W	T	F	S

```

1 // lower right
2 for (int c = col; c < board.length && r < board.length;
3     c++, r++)
4 {
5     if (board[r][c] != 'g')
6         return false;
7 }
8 return true;
9
10
11
12 public void saveBoard(char[][] board,
13 {
14     List<List<String>> allBoards)
15     String row = " ";
16     List<String> newBoard = new ArrayList<String>();
17     for (int i = 0; i < board.length; i++)
18     {
19         row = " ";
20         for (int j = 0; j < board[0].length; j++)
21         {
22             if (board[i][j] == 'g')
23                 row += 'g';
24             else
25                 row += '.';
26         }
27         newBoard.add(row);
28     }
29     allBoards.add(newBoard);
30 }
31

```

JUNE	1	2	3	4	5	6	7	8	9	10	11	12	
13	14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30										
M	T	W	T	F	S	S	M	T	W	T	F	S	

26  
Thursday

```

public void helper (char [][] board, int col,
List < List < String >> allBoards)
{
    if (col == board.length)
    {
        SaveBoard (board, allBoards);
        return;
    }

    for (int row=0; row < board.length; row++)
    {
        if (isSafe (row, col, board))
        {
            board [row] [col] = 'Q';
            helper (board, allBoards, col+1);
            board [row] [col] = '.';
        }
    }
}

```

```

public List < List < String >> solveNQueens (int n)
{
    List < List < String >> allBoards = new ArrayList <> ();
    char [][] board = new char [n] [n];

    helper (board, allBoards, 0);
    return allBoards;
}

Time complexity = O (n^n)

```

We are trying to place 'n' no. of queens in each slot of the chessboard  
 So, n queens to be tried to place in 'n' slots.

27

May

22nd Wk • 147-218

06/05  
Friday

S	T	F	S	S	S	M	T	W	T	F	S	S
9	10	11	12	13	14	15	16	17	18	19	20	21
23	24	25	26	27	28	29	30	31				

Sudoku Solver

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

Rules:-

- (i) 1-9 once in a row
- (ii) 1-9 once in a column
- (iii) 1-9 once in a grid (3x3)

1 Approach:-

✓ we shall start with  $(0, 0)$  and proceed until we reach the last cell  $(m-1, m-1)$ .

✓ we shall proceed with a tree-like structure. so, for e.g,

 $(0, 0)$  $(0, 1)$ 

1, 2, 3, 4, ... 9

so, for each cell we shall try to match which number fits and proceed to the next cell.

If, at any stage the number doesn't fit we will backtrack to the prev. cell and start from there.

JUNE	1	2	3	4	5	6	7	8	9	10	11	12
	13	14	15	16	17	18	19	20	21	22	23	24
	25	26	27	28	29	30						
M	T	W	T	F	S	S	M	T	W	T	F	S

184

28

Saturday

class SudokuSolution

→      {  
    Public boolean isSafe (char[][] board, int row, int col, int number)

    // column

    for (int i=0; i<board.length; i++)

    { if (board[i][col] == (char)(number+'0'))

        return false;

}

    // row

    for (int j=0; j<board.length; j++)

    { if (board[row][j] == (char)(number+'0'))

        return false;

}

    // grid

    int sr = 3 \* (row/3);

    int sc = 3 \* (col/3);

    for (int i=sr; i<sr+3; i++)

        for (int j=sc; j<sc+3; j++)

            if (board[i][j] == (char)(number+'0'))

                return false;

    }

    return true;

29

May

22nd Wk • 149-216

Sunday

2022

755

MAY

S	T	W	T	F	S	S	M	T	W	T	F	S
9	10	11	12	13	14	15	16	17	18	19	20	21
23	24	25	26	27	28	29	30	31				

public boolean helper (char[][] board, int row, int col)

if (row == board.length)

{ return true;

}

int nrow = 0;

int ncol = 0;

if (col == board.length - 1)

nrow = row + 1;

ncol = 0;

}

else

nrow = row;

ncol = col + 1;

if (board[nrow][ncol] != '.')

{ if (helper (board, nrow, ncol))

return true;

}

else {

// fill the place

for (int i = 1; i <= 9; i++)

{ if (!isSafe (board, nrow, ncol, i)) {

board[nrow][ncol] = (char) (i + '0');

if (helper (board, nrow, ncol)) {

return true; }

2022														
JUNIE	1	2	3	4	5	6	7	8	9	10	11	12		
MTWTFSS	13	14	15	16	17	18	19	20	21	22	23	24	25	26
MTWTFSS	27	28	29	30										

May

23rd Wk • 150-215

**30**

Monday

```

else {
    return false;
}

public void solveSudoku (char [][] board)
{
    helper (board, 0, 0);
}

```

### Few points about the code :-

1. `isSafe()` is the function where we shall check whether the number that we are going to place is located in the entire row or entire column or entire grid from that cell.
2. In the `helper()` function, there are basically 3 checks
  - (a) if we have reached the last row or not. If we have, that means all places are filled.
  - (b) if we have reached the last column, while traversing in a particular row. If yes, then we start from the next row & 1st column.
  - (c) If the place is filled or not. If it's not filled then fill the place with a number.

31

May

23rd Wk • 151-214

Tuesday

2022

157

MAY  
7 8  
21 22

## Sample Input :-

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

board =

[5, 3, 4, 2, 1, 7, 6, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]

## Sample output:

2	5	3	4	6	7	8	9	1	2
	6	7	2	1	9	5	3	4	8
3	1	9	8	3	4	2	5	6	7
	8	5	9	7	6	1	4	2	3
4	4	2	6	8	5	3	7	9	1
	2	1	3	9	2	4	8	5	6
5	9	6	1	5	3	7	2	8	4
	2	8	7	4	1	9	6	3	5
6	3	4	5	2	8	6	1	7	9

board =

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