Date: .....

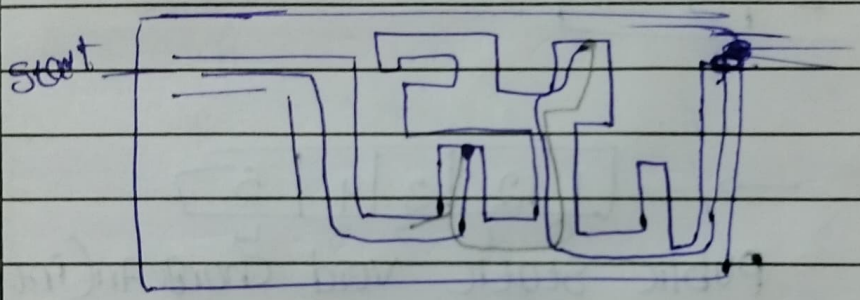
Expt. No.: .....

Page No.: .....

Back Tracking

-> Recursion
- Basic
- Divide and conquer
- Backtracking

What is back Tracking



-> It explore the solutions by going step-by-step into a choice. if choice does not lead the solution, you go back and try another choice

Types of Backtracking

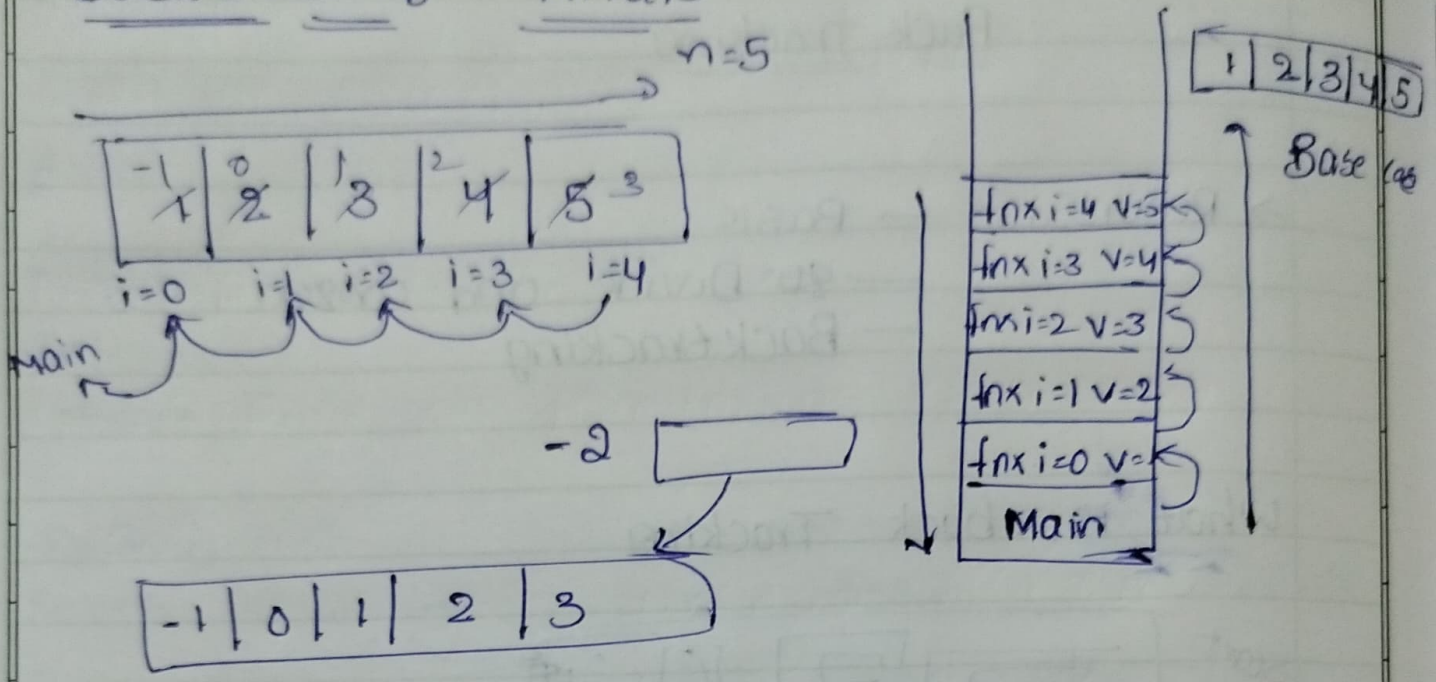
1) Decision

2) optimization

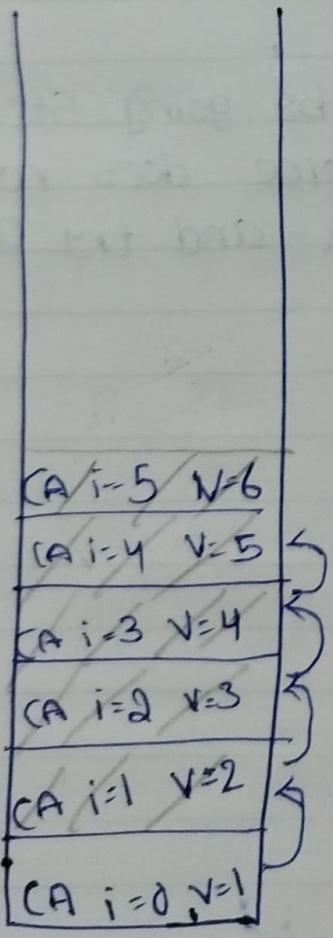
3) Enumeration

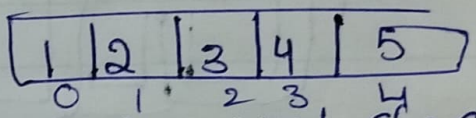
Teacher's Signature _____

Backtracking - Arrays



Call Stack






```
Public static void changeArr(int arr[],
        int i, int val) {
    // base case
    if(i == arr.length) {
        printArr(arr);
        return;
    }
    // recursion
    arr[i] = val;
    changeArr(arr, i+1, val+1);
    arr[i] = arr[i] - 2; // back tracking step
```

Date: .....

Expt. No.: .....

Page No.: .....

Find Subsets

Find & Print all Subsets of a given String

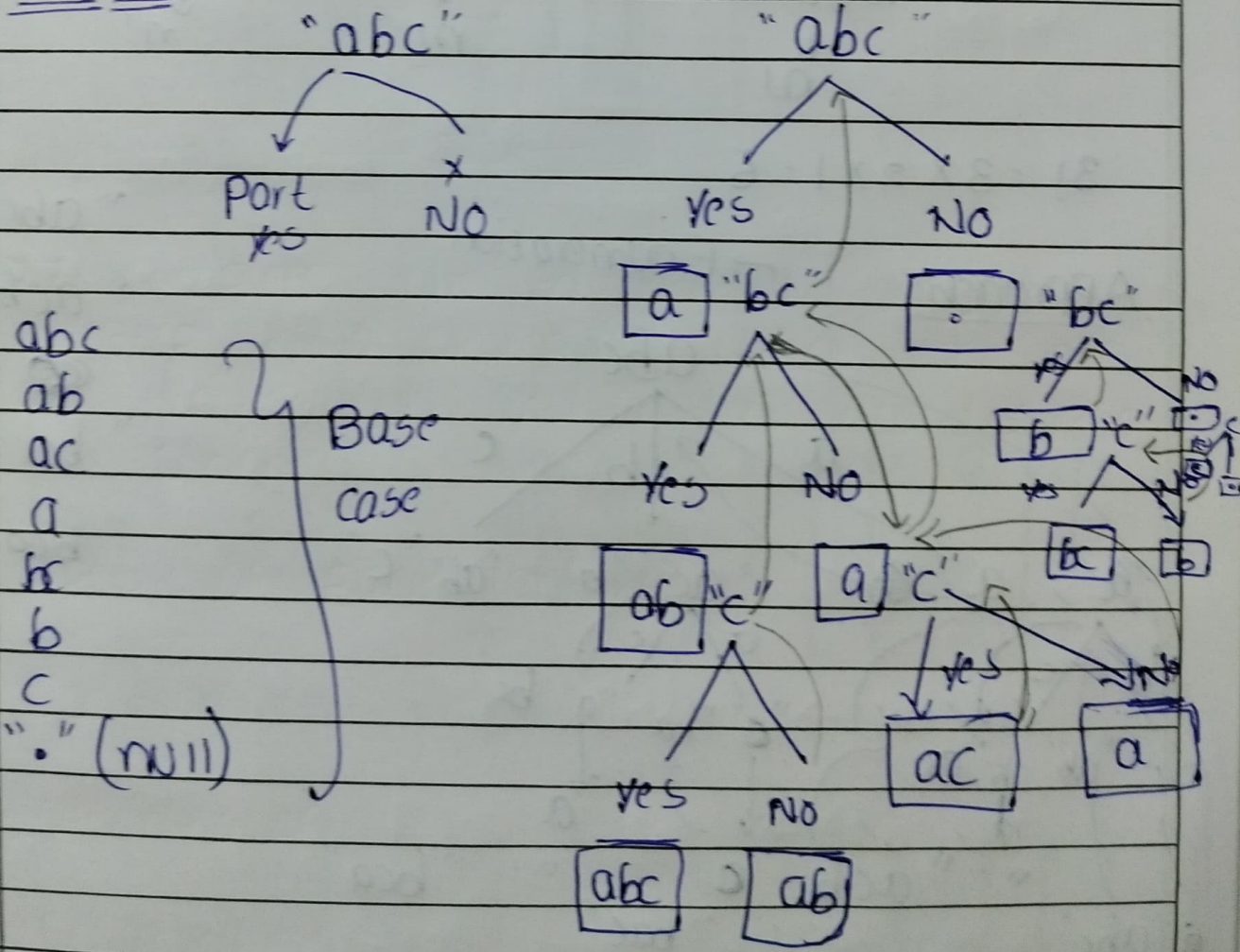
"abc"

a, b, c, ab, bc, ac, abc, "" ↓ empty set

String length n ↓ 2^n subsets

$2^3 \rightarrow 2 \times 2 \times 2 = 8$

Approach



Teacher's Signature _____

findSubsets(Str, ans) {

① ans + Str.charAt(i)

BC
→ Print

Find Permutations

Find & Print all Permutations of a String

"abc"

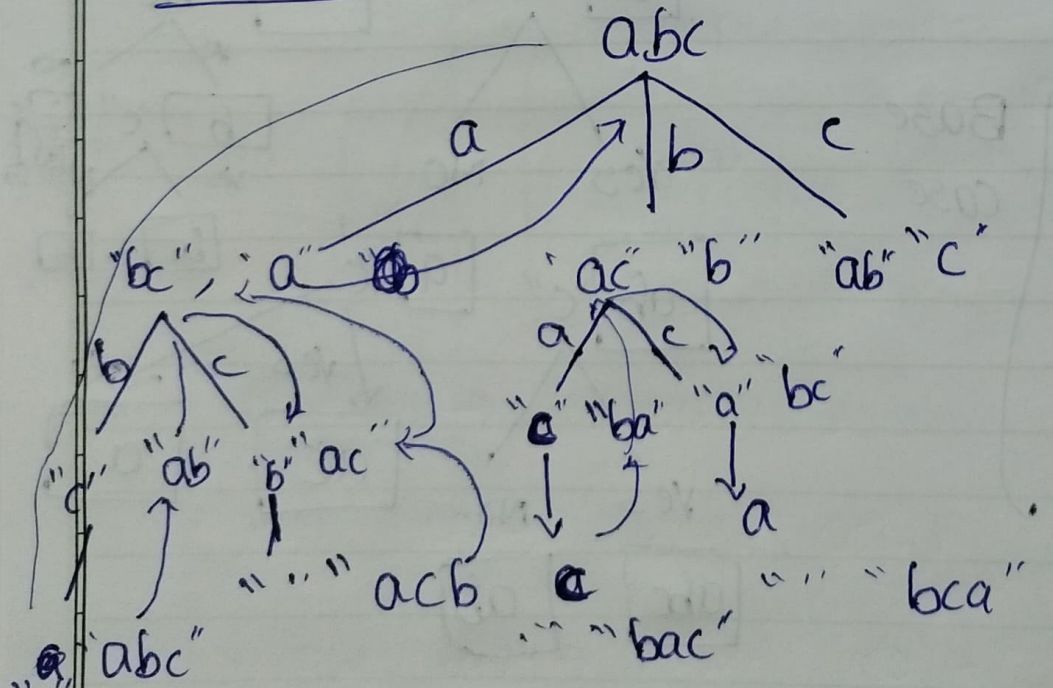
arr length n → n!

String n chars → n!

n elements → n!

3! = 3 × 2 × 1 = 6

Approach → Enumeration



"abc"
"acb"
"bac"
"bca"

Date: .....

Expt. No.: .....

Page No.: .....

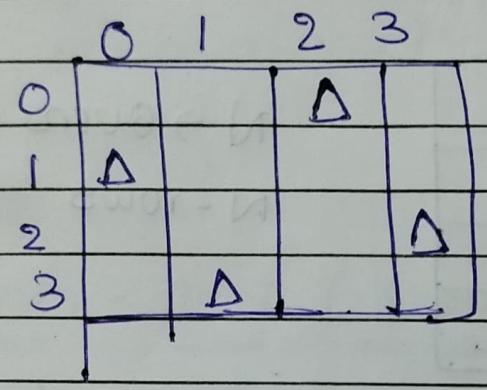
N- Queens

Place N queens on an NxN chessboard such that no 2 queens can attack each other

all solutions yes/no count sol

1 sol^n

N = 4



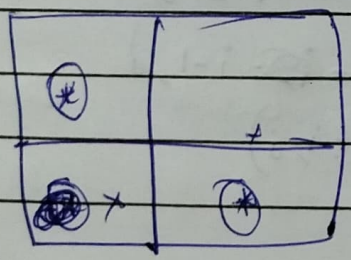
Approach

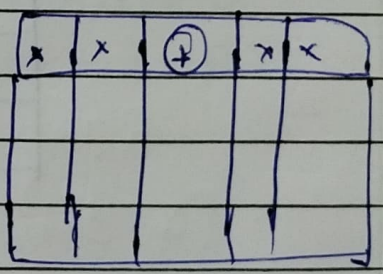
N = 2 → 4 sol

$2^2 = 4$ possible

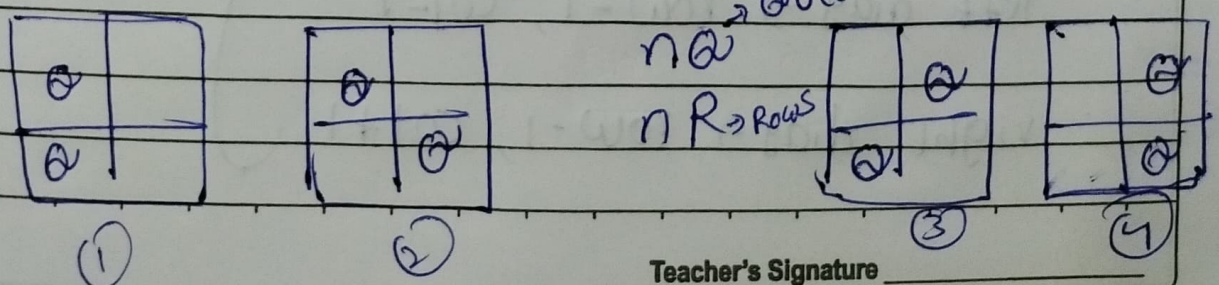
logic

vertical
horizontal
diagonal



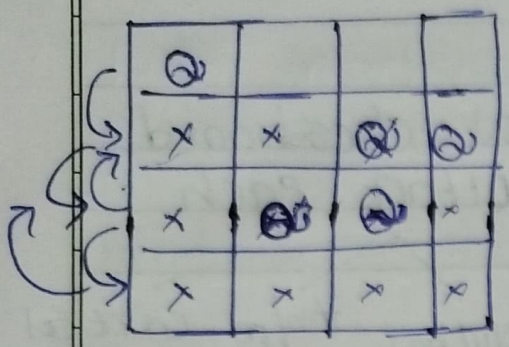


n queens n rows

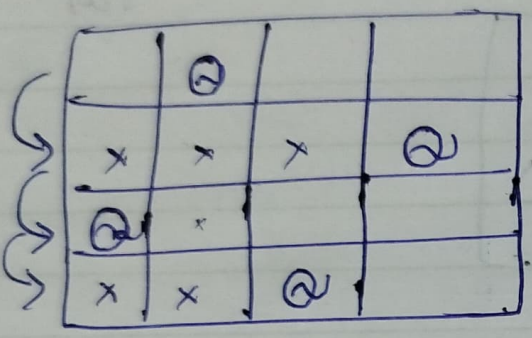


Teacher's Signature _____

N=4



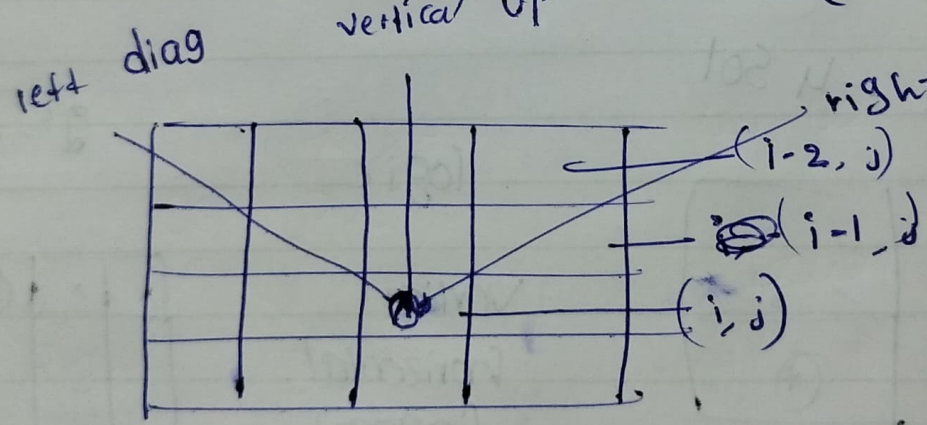
→ It's fails → change to



N → Queens

N - rows

isSafe(board, row, j)



Vertical → i = row-1, j

left diag → row-1, col-1

right diag → row-1, col+1

return false

Date: .....

Expt. No.: ..... Page No.: .....

N- Queens - Count Ways

Count total number of ways in which we can solve N Queens Problem.

static count = 0

base case

```
if(rows == board.length){
    printboard(board);
    count++;
    return;
```

N- Queens - Print 1 solution

yes → 1 Sol

No

check if problem can be solved & Print only 1 solution to N Queens Problem

```
is safe(board, row, i) {
    place
    Q(n-1)
}
```

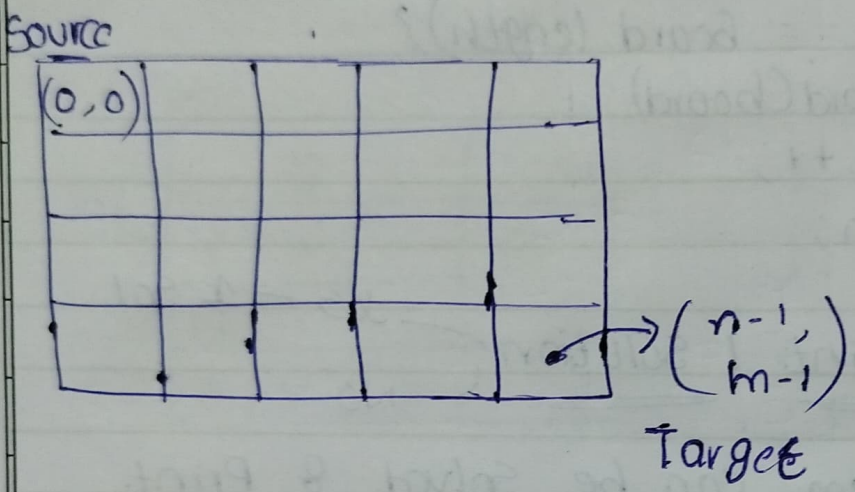
- Place
- (n-1) Q
- Unplace

True

Grid Ways

Find number of ways to reach from (0, 0) to (N-1, M-1) in a N X M Grid.

Allowed moves - right or down.



Allowed move

right ✓
down ✓

Approach

Cell (x, y) ← i, → j

Right (x, y+1)

Down (x+1, y)

w1 + w2 = total ways

Down Down

$f(x, y) = f(x+1, y) +$

$f(x, y+1)$

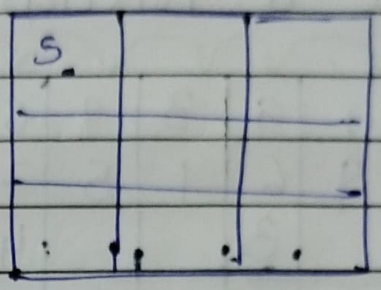
Right

Date: .....

Expt. No.: ..... Page No.: .....

Optimized way

Permutation



way (n-1) D
(m-1) R

total characters = (n-1+m-1)

DD DDRR Repeating → (n-1) D
RR RRDD (m-1) R
DRDR
DRRD

$\frac{(n-1+m-1)!}{(n-1)!(m-1)!}$ = total ways

Time comp
O(n+m)

n=3, m=3

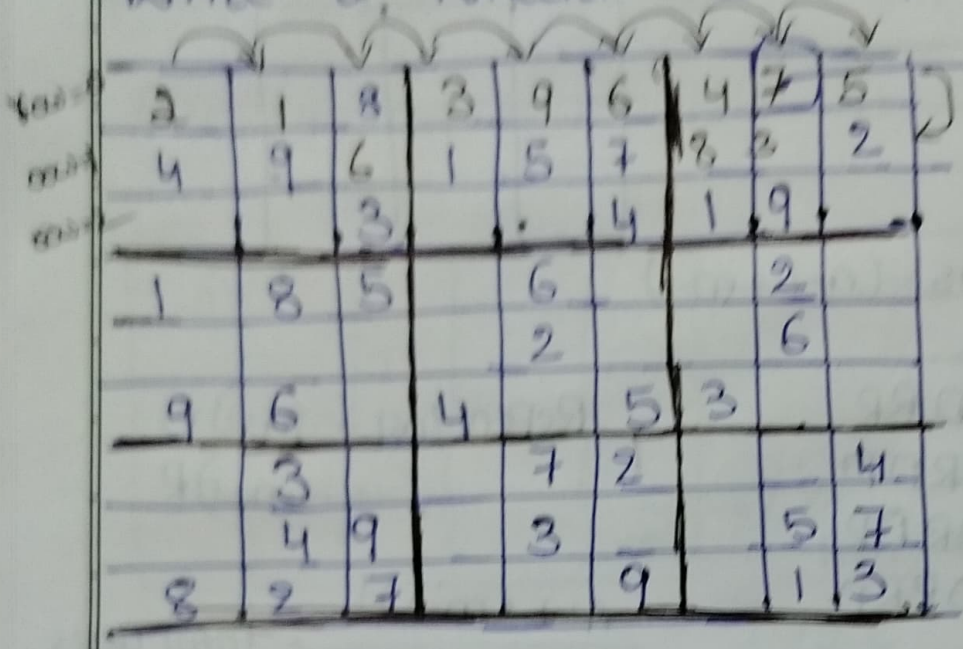
$\frac{(3-1)+(3-1)!}{2!\,2!} = \frac{(2+2)!}{2!\,2!} = \frac{4!}{2!2!} = \frac{4\times3\times\cancel{2}^{1}}{2\times1\times\cancel{2}\times1}$

= 6

Teacher's Signature _____

Sudoku

Write a function to complete a Sudoku.



Approach