

Class 76

Backtracking

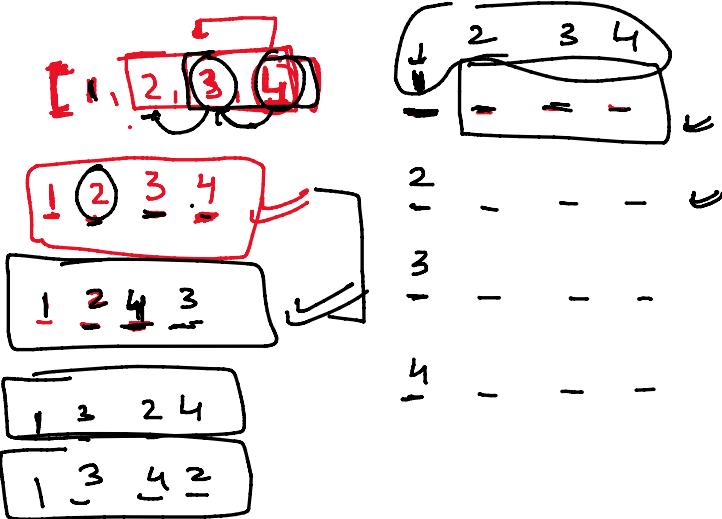
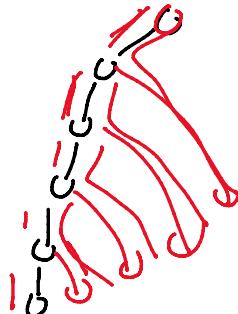
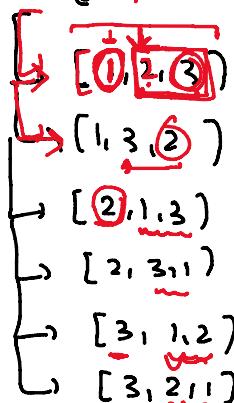
Permutation I, II

Permutation I

$$[1, 2, 3] \rightarrow n! = 3!$$

6

Recursion / Backtracking



2	-	-	-	↙
3	-	-	-	
4	-	-	-	

vector<vector<int>> Permute (vector<int> &num)

{ vector<vector<int>> ans;

temp

vector<int> temp;

int i, j, k, n;

n = num.size();

Per (ans, num, 0);

```
void per (vector<vector<int>>RanS, vector<int>nur, int id)
```

{ int n = name.size(); }

$$\int \sin(u) du = n$$

or ans.push_back(~~the~~ new);

retiree;

```
for (i = idn; i < n; i++)
```

5 Swap (name (id), name (i));

temp.push-back()

~~Peer (ans, name, idu-1);~~

~~swap (name(id), name()),~~

-keep · pop - back ()

9

[{1, 2, 3, 4}, 0]

$$\left[\left[1, \frac{2}{3}, 3, 4 \right), \emptyset \right]$$

$$\left(\left(\underline{1}, \underline{2}, \underline{3} \underline{4} \right), 2 \right)$$

$$[1, 2, 3, 4], 3$$

1,2,3,4

[1, 2, 4, 3].4

$[1, 3, 2, 4], 4$

$$[1, 3, 4, \underline{2}), 3]$$

1

{ 1, 3, 4, 2 }, 3)

Permutation II



[]

$$\begin{array}{c} [1,1,2,2] \rightarrow \\ \text{wavy line} \\ \downarrow \quad \nearrow \\ \boxed{1,1,2} \\ \boxed{1,2,1,1} \end{array}$$

$$6 \quad \left(\begin{matrix} 1, 1, 2 \end{matrix} \right)$$

$\begin{bmatrix} 2, 1, 1 \end{bmatrix}$

2. 124

3 permutations

$$\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \left[\begin{array}{c} 1,2 \\ 2,1 \end{array} \right] \left[\begin{array}{c} 1,2 \\ 1,3 \end{array} \right]$$

$$\left[\begin{array}{c} x_1 \\ x_2 \end{array} \right]$$

```
void permute (vector<int>& nums, vector<vector<int>>& ans, int i) {  
    if (i == n) {  
        ans.push_back(nums);  
        return;  
    }  
    for (int j = i; j < n; j++) {  
        swap(nums[i], nums[j]);  
        permute(nums, ans, i + 1);  
        swap(nums[i], nums[j]);  
    }  
}
```

if (idu == n)

$\Delta \text{ans} \cdot \text{Ph}(\text{news})$

return;

$m_0 P < m_1 v_0^2 / m_1$

located to a function

—
—
—

$\{ \frac{1}{2}, 1, 2 \}$ $\sqrt{10} \{ 1, 2 \}$

For(i = ider, i < n ; i++)

if (ma. count (name(i)) == 0)

↙ **swallowing), mastication**

Permes (name, ident);

Swap(new[i], new[i+1]);

$m_a \rightarrow$ 

$\text{ma}[\text{num}[i]] = 1$

$$(0-2) \quad (3-5) \quad (6-8)$$

$$\begin{pmatrix} 1_{k1} \\ 3_{13} \end{pmatrix}$$

11

$$W \leq 3$$

$$(y) = 3$$

1-9

The diagram illustrates the flow of data in a pipeline. It starts with a circled "1-9" at the top left, followed by a box containing "(5|0)". A red arrow points from this box to a circled "Sudoku Solver" box. Another red arrow points from the "Solver" box to a circled "10" at the top right. Below the "Solver" box, the number "3" is written above a red arrow pointing to a circled "1".

$$41 = \underline{3} = \bar{\underline{1}}$$

$x^1 \beta$

$$213 = \boxed{0}$$

$$1 \times 3 = 3$$

$$0 \times 5 = 0$$

(i, H₁) (

Forw^y

```
bool solve (vector<vector<char>>& board, int i, int j)
```

int nu-i;

$$\overset{=}{(i, 1+i)} - (i, i)$$

int ny = j+1;

if (i == q and j == 0)

return false;

if (j == 8)

{ nu = nu+1;
~ ny = 0; }

if (board[i][j] != '.')

{ return solve(board, nu, ny); }

~

for (k = 1; k < q; k++)

{ board[i][k] = (k + '0');

if (is_valid(board, i, j))

{ if (solve(board, nu, ny))

~

board[i][k] = '.';

~

return false;

~

bool is_valid(vector<vector<char>>& board, int i, int j)

{ if (i >= 9 || j >= 9)

for (k = 0; k < q; k++)

if (k != j and board[i][k] == board[i][j])

return false;

~

(i, j, 0)

(i, j+1)

11 column

For(k=0; k<n; k++)

 { if (k != i and board[k][j] == board[i][j])
 return false;

n

11 3x3 Matrix

int u = i / 3;
int v = j / 3;

u * = 3;
v * = 3;

For(k=u; k < u+3; k++)

 { for(z=y; z < y+3; z++)

 { if(k != i and z != j) and board[k][z] ==
 board[i][j])

 return false;

 return true;

n