

# Day-48 | LeetCode 867: Transpose of Matrix

## What is Matrix Transpose?

The transpose of a matrix is obtained by converting rows into columns. If the original matrix element is  $A[i][j]$ , then after transpose it becomes  $T[j][i]$ .

## Example

Original Matrix:  
[ [1, 2, 3],  
  [4, 5, 6] ]

Transpose Matrix:  
[ [1, 4],  
  [2, 5],  
  [3, 6] ]

## Brute Force Approach

Create a new matrix and copy each element  $A[i][j]$  to  $T[j][i]$ . This approach works for any rectangular matrix.

Time Complexity:  $O(n \times m)$   
Space Complexity:  $O(n \times m)$

## Dry Run (Brute Force)

Input:  
A = [[1, 2],  
     [3, 4]]

Steps:  
A[0][0] -> T[0][0]  
A[0][1] -> T[1][0]  
A[1][0] -> T[0][1]  
A[1][1] -> T[1][1]

Output:  
T = [[1, 3],  
     [2, 4]]

## Brute Force Code

C++:  
for(int i=0;i<n;i++)  
    for(int j=0;j<m;j++)  
        T[j][i] = A[i][j];

Java:  
for(int i=0;i<n;i++)  
    for(int j=0;j<m;j++)  
        T[j][i] = A[i][j];

Python:  
for i in range(n):  
    for j in range(m):  
        T[j][i] = A[i][j]

## Optimal Approach (In-Place)

This approach is applicable only for square matrices. Swap elements across the main diagonal without using extra space.

Time Complexity:  $O(n^2)$   
Space Complexity:  $O(1)$

## Dry Run (Optimal)

Input:

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

Swaps:

```
(0,1) <-> (1,0)  
(0,2) <-> (2,0)  
(1,2) <-> (2,1)
```

Output:

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

## Optimal Code

```
C++:  
for(int i=0;i<n;i++)  
    for(int j=i+1;j<n;j++)  
        swap(A[i][j], A[j][i]);  
  
Java:  
for(int i=0;i<n;i++)  
    for(int j=i+1;j<n;j++){  
        int t=A[i][j];  
        A[i][j]=A[j][i];  
        A[j][i]=t;  
    }  
  
Python:  
for i in range(n):  
    for j in range(i+1, n):  
        A[i][j], A[j][i] = A[j][i], A[i][j]
```

## Interview Summary

Brute Force: Works for any matrix, uses extra space  
Optimal: Works only for square matrix, no extra space