

Determinant of a Matrix | Explanation

What is the Determinant of a Matrix?

Determinant of a Matrix is a special number that is defined only for square matrices (matrices which have the same number of rows and columns). The determinant is used at many places in calculus and other concepts related to algebra, it actually represents the matrix in term of a real number which can be used in solving a system of linear equations and finding the inverse of a matrix.

How to calculate Determinant?

The value of the determinant of a matrix can be calculated by the following procedure:

*For each element of the first row or first column get **cofactor** of those elements and then multiply the element with the determinant of the corresponding cofactor, and finally add them with alternate signs. As a base case, the value of the determinant of a 1*1 matrix is the single value itself.*

Cofactor of an element, is a matrix which we can get by removing row and column of that element from that matrix.

Examples



Dash



All



Articles



Videos



Problems



Quiz

<< Prev

Next >>





Dash



All



Articles



Videos



Problems



Quiz



Determinant of 2 x 2 Matrix:

$$A = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

$$|A| = (ad - bc)$$



Remember by this -

$$\begin{matrix} a & b \\ c & d \end{matrix} \begin{matrix} \diagdown \\ \diagup \end{matrix}$$

Determinant of 3 x 3 Matrix:

$$A = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

$$|A| = a(ei - fh) - b(di - gf) + c(dh - eg)$$

In terms of Cofactor:

$$\begin{bmatrix} a & e & f \\ x & h & i \end{bmatrix} - \begin{bmatrix} b & f \\ x & i \end{bmatrix} + \begin{bmatrix} d & e \\ g & h \end{bmatrix} \begin{matrix} c \\ x \end{matrix}$$

Calculating Determinant of an NxN Matrix

To calculate determinant of an NxN matrix, we will create two functions namely:

- **determinantOfMatrix()**: This function will accept two parameters, the matrix and its dimension and will return the value of the determinant of the given matrix.
- **getCofactor()**: This function will return cofactor of the given element in the first row of a given matrix.

The step-by-step approach is:

- Inside the **determinantOfMatrix()** function, run a loop from 0 to N for all elements of the first row.
- For every element in the first row, find its cofactor matrix using the **getCofactor()** function.
- Then, multiply the element with the determinant of the corresponding cofactor, and finally add them with alternate signs.
- To find the determinant of the current co-factor, call the **determinantOfMatrix()** function recursively, and pass the cofactor matrix and its dimension as parameters.

Implementation of the `getDeterminant()` function:

CPP

```
/* Recursive function for finding determinant of matrix.
   n is current dimension of mat[][]. */
int determinantOfMatrix(int mat[N][N], int n)
{
    int D = 0; // Initialize result

    // Base case : if matrix contains single element
    if (n == 1)
        return mat[0][0];

    int temp[N][N]; // To store cofactors

    int sign = 1; // To store sign multiplier

    // Iterate for each element of first row
    for (int f = 0; f < n; f++)
    {
        // Getting Cofactor of mat[0][f]
        getCofactor(mat, temp, 0, f, n);
        D += sign * mat[0][f] * determinantOfMatrix(temp, n - 1);

        // terms are to be added with alternate sign
        sign = -sign;
    }
}
```



Dash



All



Articles



Videos



Problems



Quiz





Dash



Courses

Tutorials

Jobs

Practice

Upcoming
Contests

All



Articles



Videos



Problems



Quiz



```
}


return D;


}
```


```
// Function to get cofactor of mat[p][q] in temp[][]. n is current
// dimension of mat[][]
void getCofactor(int mat[N][N], int temp[N][N], int p, int q, int n)
{
    int i = 0, j = 0;


    // Looping for each element of the matrix
    for (int row = 0; row < n; row++)
    {
        for (int col = 0; col < n; col++)
        {
            // Copying into temporary matrix only those element
            // which are not in given row and column
            if (row != p && col != q)
            {
                temp[i][j++] = mat[row][col];


                // Row is filled, so increase row index and
                // reset col index
                if (j == n - 1)
                    i++;
            }
        }
    }
}
```



Dash



All



Articles


Videos


Problems


Quiz





```
{  
    j = 0;  
    i++;  
}  
}  
}  
}  
}
```

[Mark as Read](#) [Report An Issue](#)

If you are facing any issue on this page. Please let us know.