DSA SERIES

- Learn Coding

Topic to be Covered today

Merge sorting

LETS START TODAY'S LECTURE

Merge sorting

- Sorting algorithm
- This algorithm works on the concept of the divide and conquer.

Divide and Conquer is an algorithm design paradigm where a problem is:

- **Divided** into smaller subproblems (usually of the same type),
- Each subproblem is **conquered** (solved) independently,
- The results of subproblems are then **combined** to form the solution to the original problem.

The same approach we will use to sort our array .

Algorithm :

- Split the array into two halves repeatedly .
- Sort the split array .
- Finally combine them .

Code :

```
#include <iostream>
#include <vector>
using namespace std;

void merge(vector<int> &arr, int left, int mid, int right)
{
    vector<int> temp;
    int i = left;
    int j = mid + 1;

    while (i <= mid && j <= right)
    {
        if (arr[i] <= arr[j])
        {
            temp.push_back(arr[i++]);
        }
        else
        {
            temp.push_back(arr[j++]);
        }
    }
    while (i <= mid)
    {
        temp.push_back(arr[i++]);
    }

    while (j <= right)
    {
        temp.push_back(arr[j++]);
    }

    for (int i = left; i <= right; i++)
    {
        arr[i] = temp[i - left];
    }
}
```

```
void mergeSort(vector<int> &arr, int left, int right)
{
    if (left >= right)
        return;

    int mid = (left + right) / 2;
    mergeSort(arr, left, mid);
    mergeSort(arr, mid + 1, right);

    merge(arr, left, mid, right);
}
```

```
int main()
{
    vector<int> arr = {8, 4, 5, 2, 9, 1};
    int n = arr.size();

    cout << "Priting the original array : ";
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
    mergeSort(arr, 0, n - 1);

    cout << "Priting the  array after sorting: ";
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}
```

Learn coding

THANK YOU