27/5/21

Q → Merge Sort

```
#include <stdio.h>
#include <time.h>
void merge (int a[], int l, int m, int r){
    int i, j, k, c[10000];
    i = k = l;
    j = mid + 1;
    while (i <= mid && j <= r) {
        if (a[i] < a[j]) {
            c[k] = a[i];
            ++k;
            ++i; }
        else { c[k] = a[j];
            ++k, ++j; }
    }
    if (i > mid) {
        while (j <= r) {
            c[k] = a[j];
            ++k, ++j; }
    }
    if (j > r) {
        while (i <= mid) {
            c[k] = a[i];
            ++k, ++i; }
    }
```

```
    for (i=l; i<=r; i++)
        a[i] = c[i];
}
void mergeSort (int a[], int l, int r){
    int mid;
    if (l<r){
        mid = (l+r)/2;
        mergeSort (a, l, mid);
        mergeSort (a, mid+1, r);
        merge (a, l, mid, r);
    }
}
int main () {
    int a[10000], n, i;
    printf ("Enter no. of elements : ");
    scanf ("%d", &n);
    printf ("Enter array elements");
    for (i=0; i<n; i++)
        a[i] = rand ();
    clock_t begin = clock ();
    mergeSort (a, 0, n-1);
    clock_t end = clock ();
    double time = 0.0;
    time = (double) (end - begin) / CLOCKS_PER_SEC;
    printf ("\n Sorted Array :- ");
```

```
for (i = 0 ; i < n ; i++)
    printf("%d ", a[i]);
printf("Time : %f", n - time);
return 0;
}
```