Merge Sort

```
#include<stdio.h>
#include<time.h>
void split(int n, int a[])
{
    int mid,i,j
    if (n<2)
        return
    mid = n/2;
    int left[mid]
    int right[n-mid]
    for(i=0; i<mid; i++)
        left[i] = a[i]
    for(j=mid; j<n; j++)
        right[j-mid] = a[j]

    split( sizeof(left)/sizeof(int), left)
    split ( sizeof(right)/sizeof(int), right)
mergesort(left, right, a, sizeof(left)/sizeof(int)
                , sizeof(right)/sizeof
                (int))

}
```

```
void mergsort(int left[], int right[], int a
{
    int i=j=k=0
    while (i<nl && j<nr)
    {
        if (left[i]<right[i])
        {
            a[k] = left[i]
            i++;
        }
        else
        {
            a[k] = right[j]
            j++;
        }
        k++
    }
    while (i<nl)
    {
        a[k] = left[i]
        i++
        k++
    }
    while (i<nr)
    {
        a[k] = right[j]
        j++
        k++
    }
}
```

, int nl, int nr)

int main()
{
clock_t t;
int n;
printf("Size\n");
scanf("%d", &n);
int a[n];
printf("Elements\n");
for(int i=0; i<n; i++)
scanf("%d", &a[i]);
t=clock();
split(n, a);
t = clock() - t;
double time_taken = ((double)t)/CLOCKS_PER_SEC;
printf("%f\n", time_taken);
printf("order is\n");
for(int i=0; i<n; i++)
printf("%d\t", a[i]);

}