```
#include<stdlib.h>
#include<stdio.h>
#include <time.h>

/*using merge sort for sorting by weight*/
void merge(int arr[][2], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 =  r - m;

    int L[n1][2], R[n2][2];

    for (i = 0; i < n1; i++) {
        L[i][0] = arr[l + i][0];
        L[i][1] = arr[l + i][1];
        }
    for (j = 0; j < n2; j++) {
        R[j][0] = arr[m + 1+ j][0];
        R[j][1] = arr[m + 1+ j][1];
    }

    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2)
    {
        if (L[i][0] < R[j][0])  {
            arr[k][0] = L[i][0];
            arr[k][1] = L[i][1];
            i++;
        }
        else if (L[i][0] == R[j][0]) {
            if (L[i][1] >= R[j][1]) {
                arr[k][0] = L[i][0];
                arr[k][1] = L[i][1];
                i++;
            } else {
                arr[k][0] = R[j][0];
                arr[k][1] = R[j][1];
                j++;
            }
        }
        else {
            arr[k][0] = R[j][0];
            arr[k][1] = R[j][1];
            j++;
        }
        k++;
    }
    while (i < n1)
    {
        arr[k][0] = L[i][0];
        arr[k][1] = L[i][1];
        i++;
        k++;
    }
    while (j < n2)
    {
        arr[k][0] = R[j][0];
        arr[k][1] = R[j][1];
        j++;
        k++;
    }
}
```

```
void mergeSort(int arr[][2], int l, int r)
{
    if (l < r)
    {
        int m = l+(r-l)/2;
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);
        merge(arr, l, m, r);
    }
}


int main()
{
    clock_t start_t, end_t, total_t;
    printf("Enter number of elements \n");
    int n;
    scanf("%d",&n);
    int i=0;
    int a[n],w[n],temp[n][2];
    printf("Enter elements \n");
    for(i=0;i<n;i++) {
        scanf("%d",&a[i]);
        temp[i][1] = a[i];
    }
    printf("Enter weight of elements \n");
    for(i=0;i<n;i++) {
        scanf("%d",&w[i]);
        temp[i][0] = w[i];
    }
    //for time calculation
    start_t = clock();

    mergeSort(temp, 0,n - 1);

    //time end here


    printf("Sorted Array: \n");
    for(i=0;i<n;i++) {
        printf("%d %d\n",temp[i][1],temp[i][0]);
    }
    end_t = clock();
    total_t = ((double)(end_t - start_t) / CLOCKS_PER_SEC)*CLOCKS_PER_SEC;
    printf("Total time taken by CPU: %f\n", (double)total_t/(double)CLOCKS_PER_SEC
);
    printf("Exiting of the program...\n");
}
```