```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <omp.h>

#define MAX_SIZE 10000

void generate_list(int * x, int n) {
  int i;
  for (i = 0; i < n; i++)
   x[i] = rand() % n;
}

void merge(int * X, int n, int * tmp) {
  int i = 0;
  int j = n/2;
  int ti = 0;
  while (i<n/2 && j<n) {
    if (X[i] < X[j]) {
      tmp[ti] = X[i];
      ti++; i++;
    } else {
      tmp[ti] = X[j];
      ti++;
        j++;
    }
  }
  while (i<n/2) {
    tmp[ti] = X[i];
        ti++;
        i++;
  }
    while (j<n) {
      tmp[ti] = X[j];
      ti++;
        j++;
  }
  memcpy(X, tmp, n*sizeof(int));

}

void mergesort(int * X, int n, int * tmp)
{
  if (n < 2) return;

  #pragma omp task firstprivate (X, n, tmp)
  mergesort(X, n/2, tmp);

  #pragma omp task firstprivate (X, n, tmp)
  mergesort(X+(n/2), n-(n/2), tmp);

  #pragma omp taskwait
  merge(X, n, tmp);
}

int main()
{
  int n = MAX_SIZE;
  double start, stop;

  int data[MAX_SIZE], tmp[MAX_SIZE];

  generate_list(data, n);
  start = omp_get_wtime();
  #pragma omp parallel
  {
    #pragma omp single
    mergesort(data, n, tmp);
  }
  stop = omp_get_wtime();
  printf("Time: %g\n",stop-start);
}
```

serial

```
> gcc merge.c -fopenmp -o merge
> ./merge
Time: 0.00509333
```

parallel

```
> gcc merge.c -fopenmp -o merge
> ./merge
Time: 0.0591125
```