Merge Sort using multithreading

The algorithm divides the data set into equal-sized subsets. The number of subsets equals the number of threads used. The individual threads use the merge sort code to sort the data individually. After all the threads complete the sorting the data is merged back into the bigger array.

Bucket Sort using multithreading

The bucket sort algorithm uses presorted arrays, known as multiset in c++. Similar to the merge sort algorithm, the data is divided into equal sets, where the total number of sets equal the number of threads used. Each thread inserts the elements into buckets which are sorted. The sorted buckets can then be copied into the bigger array.

The user can choose on runtime if the it wants to use merge sort or the bucket sort algorithm.

Compilation instructions

make : to compile the code

Make clean : to clean the executables

Execution instructions

./mysort --name : prints name

./mysort <inputfile> -t <number of threads> -o <outputfile> --alg=<fj,bucket>