MergeSort

Iterative vs Recursive

MergeSort as the possibility of being either recursive or iterative. The program given explores the difference in speeds for both merge types, for different types of list in increasing amounts.

The program comes with 3 possible arguments

-I Tells which increment type to use, Default is 4

type 1 : ++value

type 2 : value = value \* 2

type 3 : value = value \* value

type 4 : changes in order to always have only 100 points

-sA is used to change the maximum stress value or the largest array the program will have to sort, default is 100000.

be careful : when changing the increment type you should also change this accordingly

-p is used to set if the programs opens the .txt windows or not after it is done running, default is true

Use like so

./Program -help

./Program -I 4 -sA 10000 -p true

# Comparisons

## The Main differences

The main differences between the is really the space they take, the iterative takes much less space then the recursive since it only uses a for loop. The recursive on the other hand calls itself many times, leading to more memory space usage.

## The graphs

When the program is done running you should get two .txt files pop up with a table, if option 4 is chosen, and graphs of each type of data that was sorted. For example :

A graph on a screen

AI-generated content may be incorrect.

This is the recursion for merge sort with normal unsorted array.

Now lets see which one is faster.

Please run the program yourself and see if you have similar results

## Sorted

The real difference is here, when implementing the check if sorted we can simply rule out having to do the merge sort if its already sorted for the iterative which then gives us the following graphs :

Recursive :  
A graph on a screen

AI-generated content may be incorrect.

Iterative :

A graph showing a graph

AI-generated content may be incorrect.

As we can see the iterative is MUCH faster more then 50 times faster in fact. Unfortunately, this is where the separation really finishes. And this is only because it checks if its sorted, which it shouldn’t.

## Randomly assorted

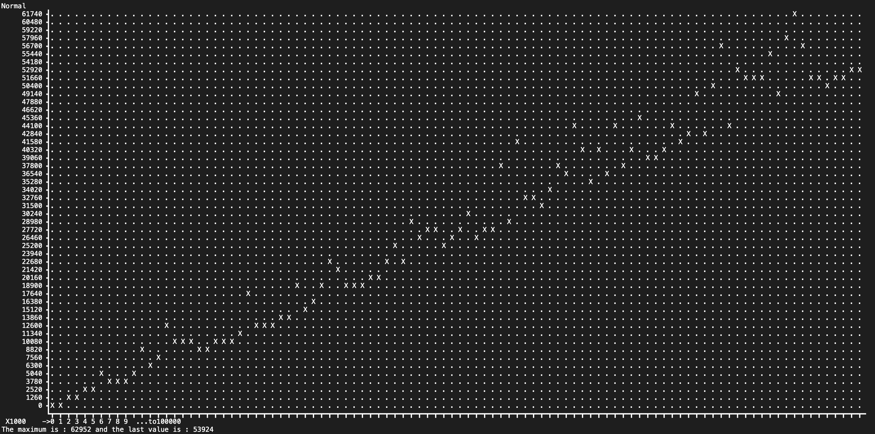
Here we can see that there isn’t much of a difference

Recursive :

A graph of a graph

AI-generated content may be incorrect.

Iterative :



We can see that there isn’t much of a difference although it does seem like the iterative one is faster but honestly not that much.

## Conclusion

The rest is mostly the same by running the program all of the graphs should be given. The files should be named the following:

recursive.txt

itterator.txt