Sorts

A Demo of Sort Algorithm Efficiency

This is re-write or modification of the SORTS VB program previously uploaded to Compuserve's MSBASIC forum.

The actual sorting algorithms *have not been changed* - only the user interface has been modified, with the addition of writing a "results" file that can be directly imported (OPEN as .TXT) into Excel or another spreadsheet. The text file was added to allow the user to import the time, iterations, and number of swaps into a spreadsheet, for further analysis - or simply to allow generating a hardcopy for future reference. Too, the program was modified so that each algorithm uses the same series of 100 random numbers, to preclude any variance in the random numbers from biasing the results. Finally, the program has also been modified so that it will 'share' with the other programs - meaning that now you can iconize it, switch over to some other program, or whatever (for you new VB programmers, include PLENTY of [something]=DoEvents() scattered around in your programs, particularly in loops; those are how VB lets the rest of Windows operate).

Since the original didn't include any description of what the program was doing, or what the activities should be, here they are:

The program is simply meant to use data (a set of 100 random numbers) to test different sorting algorithms. The data is initially displayed as discrete points on a graph; as the sort progresses, you will see a diagonal line - from top left to bottom right - start to develop. Too, as the sort advances, you'll be able to see the elapsed time, number of iterations (cycles through) the sort, and the number of times two values have been swapped. Once the sorts are done, the program will proceed directly to the next algorithm (they're now in alphabetic order). At any point during the program's operation, you can click on the EXIT button, and the program will terminate (!!) after writing a text file containing the results of all the **completed** algorithms.

If you let the program run to completion (suggested, but it may take a while on slower machines), when you exit, you'll get the file C:\sort_lst.txt which you can either view with any text editor, or bring into a spreadsheet. You might want to bring it into a spreadsheet so that you can do some math on the results, and derive iterations/second, or swaps/iteration, or whatever you want. Note that all times given in the file are in SECONDS (makes the math easier), while the program display is in HH:MM:SS.

Many thanks to the original uploader for taking the time to code all these sorting algorithms, and making the source code available. I can only hope that the modifications and changes I've made aren't *too* disagreeable!

A final NOTE:

You might want to run the program several times (perhaps as many as 100), so as to get enough data to have a proper statistical base - one run might be a bit better than another. Or, you might want to consider modifying the program to generate random-length strings of random characters, to see how they do on text. Just remember, with the current version of the program, each time you exit the program, it **WILL** overwrite the previous copy of SORT_LST.TXT; so you need to remember to either rename the file each time, or load it into your spreadsheet.

Have fun!