EXOSORT FOR STRINGS

EXOSORT stands for sorting by exiting. It comes from the Greek Exodus + Sort. The idea is to sort using most significant digit radix sorts however exists the information to process. It is similar to a program on the web by David B. Ring at **http://www.devx.com/vb2themax/Tip/19471** although **EXOSORT** does not sort in place.

This type of sort which we both use is the fastest sort by using memory. The more memory available to a certain extent the faster it sorts**. EXOSORT** looks up every character once, for every string, and when there is a difference it stops the sort making **EXOSORT** the fastest sorting routine. In compare sort such as Quicksort, the compare instruction compares every character in the string and gives an execution when there is a difference in the characters, and then the sort will go back and compare that string again to another string. Thus, more time is spent in sorting. Obviously, if there is no available memory, a compare sort will be better suited than **EXOSORT**. It turns out looking up a character plus an addition is faster than comparing one character to another with an instruction. So, even in a compare sort, if it does not go back and compare that string to another string **EXOSORT** would be faster since **EXOSORT** does not compare. The chart shown on this webpage demonstrates the result for using 5N extra memory where N is the number of strings in a table and equal 6 character string.

If more information is needed on **EXOSORT** please email **farid@facsystems.com**

PS. Also, there is an **EXOSORT** for numbers and it is similar to a spreadsort by Steven J. Ross but with no buckets. The sort is faster than treating the numbers as strings when utilizing a string sort and is faster than spreadsort, as it does not use a compare instruction.

Farid A. Chouery, P.E., S.E.

Structural, Civil, Electrical Engineer
President/CEO FAC Systems Inc.