CS 6301.{013,016}: Implementation of advanced DS&A

Project 1 (Type 2: Empirical study of merge sort)

Tue, Aug 26

Version 2.0:

\* Replaced [Sort.java](http://www.utdallas.edu/~rbk/teach/2014f/Sort.java) by [BetterSort.java](http://www.utdallas.edu/~rbk/teach/2014f/BetterSort.java).

\* Added additional sizes for testing.

\* Naming convention for files: Name your file using the netid of one of the

group members as the prefix of your file (so, I would name it rbkSort.java).

Only one member per group needs to upload the file. Make sure that names of

all group members appears in your source files and the report.

\* [Old description](http://www.utdallas.edu/~rbk/teach/now/p1-6301-2014f.html)

Deadlines: 2 AM on Monday: Sep 8 (1st), Sep 15 (2nd), Sep 29 (3rd).

Download code for merge sort from <http://www.utdallas.edu/~rbk/teach/2014f/BetterSort.java> and modify the code to implement the following 3 versions of merge sort discussed in class. Translate to C++ or Python if you want to work in those languages. Compare their running times for the following values of n: 8192, 65536, 1048576, 8388608. Add additional code as needed to find the running time of the programs. Run your program on the following values of n, if your machine has enough hardware: 8388608 16777216 33554432 67108864 134217728 268435456 536870912 1073741824

1. Allocate dynamic memory for L and R within Merge.
2. Create a single auxiliary array B in main and pass it to MergeSort and Merge. In each instance of Merge, data is copied from A to B and merged back into A.
3. Create a single auxiliary array B in main and pass it to MergeSort and Merge. Data alternates between A and B. Extra credit for code that works correctly for all values of n (not just powers of 2).

Specification: Program takes one integer on the command line (n), creates an array of that size and populates it with numbers in reverse sorted order. It sorts the array using merge sort and verifies whether the array is in sorted order. It prints just one line of output indicating whether the algorithm succeeded in sorting the array.

Upload your code on elearning, along with a report that gives the running times (in msec) of the 3 algorithms for the 4 different values of n (in tabular form). Explain if your code is eligible for extra credit and how you implemented it to run for all values of n.