

Finding the Lost Sorts: Teaching Empirical Algorithm Analysis*

Nifty Assignment

Dennis Bouvier[†]

Department of Computer and Cyber Sciences
United States Air Force Academy
USAFA, CO 80840

djb@acm.org

“Lost Sorts” an activity in which students identify the sorting algorithm by runtime data. The name “Lost Sort” comes from the backstory that runtime data were collected on sorting algorithms, but the experimenter lost the algorithm identifiers. Rather than re-run the timing, the student is tasked with identifying the algorithms by doing the analysis of run times.

The implementation of the activity provides each student a different set of data and automatic assessment of their responses. The Lost Sort activity has been used in a CS2 course with good success in learning while not burdening the instructor in grading a multitude of individual assignments.

The sorting algorithms included in this lesson are selection, insertion, bubble, merge, and quicksort. The bubble sort implementation is a one-way pass that does not detect early termination. The implementation of quicksort uses a fixed position pivot selection. Three runs of data were collected for each algorithm for each of four input conditions: in-order, reverse-order, random-order, and input with repeated values

To assist students in identifying the algorithms students answer questions in a online quiz in our LMS. The quiz questions are: their assigned Data-ID (that was generated from the hash of their email address) and two sets of ‘questions’ for each of the six sets of data. The first set of question is nine true/false

*Copyright is held by the author/owner.

[†]The views expressed are those of the author and do not reflect the official policy or position of the US Air Force, Department of Defense or the US Government. Approved for public release: distribution unlimited. PA# USAFA-DF-2025-784

statements (list below), the second set is one question that asks the identity of the sorting algorithm.

- the worst case runtime is order N
- the order of the runtime for a randomly ordered list is N-squared
- the sorting algorithm is adaptive
- the order of the runtime for a randomly ordered list is $N \log N$
- the order of the runtime for a reverse ordered list is $N \log N$
- the order of the runtime for a reverse ordered list is N-squared
- the worst case runtime is order N-squared
- the order of the runtime for a reverse ordered list is N
- the worst case runtime is order $N \log N$

Prior to this assignment, students have learned about sorting algorithms. An in-class presentation demonstrating the analysis runtime data in Excel™ introduces the activity. Students download a copy of the activity spreadsheet. Initially, no data are visible in the spreadsheet. The student is required to enter an email address in the first sheet to select the data for that student. A hash computed from the email address selects the data for that student, thus each student has a unique set of data for the assignment. That allows students to collaborate without an opportunity to copy/paste answers.

The data for the assignment appears in six other sheets. The presentation of the data in the spreadsheet makes it easy for the student to do the analysis of the data assigned to them. The sheets are locked and password protected so as not to reveal the answers. Because the sheets with the data are locked, students must copy/paste the data to a new sheet to perform the analysis; however, this prevents the student from accidentally altering the data.

The Lost Sort activity depends only on students having been exposed to the sorting algorithms of the activity. It should be reasonably easy to adopt this activity as is. Versions of the “Lost Sort” PowerPoint™ presentation and “Lost Sort” Excel™ spreadsheet can be found on Zenodo. The spreadsheet includes instructions for customizations, including assigning a salt value used in the randomization.