INFO 6205 Fall 2021
Team Project

Abstract

Your task is to implement MSD radix sort for a natural language which uses Unicode characters. You may choose your own language or (Simplified) Chinese. Additionally, you will complete a literature survey of relevant papers and you will compare your method with Timsort, Dual-pivot Quicksort, Huskysort, and LSD radix sort.

Requirements

Report (15)

You will write a report which summarizes your overall findings and recommendations—and shows, graphically, your main conclusions. This report will typically be two to five pages in length, depending on the number and size of the graphics

Benchmark and Results (20)

You will use the *SortBenchmark* class from the class repository, or you may adapt the *HuskySortBenchmark* class from the *HuskySort* repository (https://github.com/rchillyard/The-repository-formerly-known-as). [Don't ask me why that is the title. I honestly don't remember.] Or you can just go back to your work on assignment 2. You should benchmark the results of all methods (see the list in the abstract) for 250k, 500k and 1M, 2M, 4M names at least. Your sorting will be judged based on the 1M names. I am providing a file of 1M randomly ordered names which you should use as input (for the 2M, 4M benchmarks, just build the list two or four times). The conventional order for Chinese is according to the English order of the Pinyin.

Paper and Literature Survey (20)

You must write a paper which explains the work that you have done in some detail and describes the work of at least two related technical papers, i.e., on the subject of MSD Radix sort, MSD Radix Exchange sort, or other technique you believe is relevant. I would recommend finding four such papers if yours is a three-person team. Your paper should be at least three pages long and up to five pages. You must format it much as if it was to be published (don't convert to LaTex, though). You may simply copy the abstract from this paper into your report.

Code (20)

You must push all of your code (as well as the repo to your repository and submit the name of that repository. Include your unit tests of course. Your README must elaborate on the new modules and on any changes that you have made to existing code. Most of your code should be in new modules.

Unit tests (12)

It is very important that all of your code (whether new or edited) is properly unit-tested. This is an area where, traditionally, students have lost points.

On Time (13)

Your Github repo must be timestamped no later than 11:59:59 pm, December 5th, 2021. Any project that is late will lose at least 5 points. The penalty increases significantly in the days that follow.

Plagiarism

Please note that plagiarism will invoke serious deductions. The work must be the work of your own team. Wherever you use the work of others, you must attribute it (including any reference to the HuskySort paper). You do not need to attribute the code you get from the class repository or the HuskySort repository.

Useful Links

Unicode: https://home.unicode.org

Other Notes

Please note that Huskysort does not currently sort Chinese characters in the correct order (English order of Pinyin). Expect to find an update of the code within a week or so.