CS591 A1 Spring 2020 - Research Project

Title: *Build a sortedness benchmark*

Background: Sorting algorithm behave differently according to the physical order of the input to be sorted. Similarly, indexing data structures that add *orderness* to the inserted values at the cost of insertion are expected to behave differently for different physical order of the input keys.

Objective: This project aims to develop a new benchmark that can be used to measure latency as the sortedness of the input is varied. Even though there is no single metric for sortedness defined yet, in this project you are expected to clearly define perturbations of sorted data collections to clearly show how performance is affected as sortedness changes.

The workflow of the project is as follows:

(a) Review the existing metrics that are proposed to quantify the orderliness or pre-sortedness of a dataset [1, 2]
(b) Given a sorted collection, design a family of increasingly "less sorted" collections.
(c) Measure the runtime of in-memory sorting algorithms with these collections.
(d) Implement a simple API that inserts these collections to simple data structures and measure insert latency and read latency
(e) Measure read latency with variable sortedness using zonemaps

[1] Heikki Mannila. **Measures of Presortedness and Optimal Sorting Algorithms**. IEEE Trans. Computers 34(4): 318-325 (1985)

[2] Sagi Ben-Moshe, Yaron Kanza, Eldar Fischer, Arie Matsliah, Mani Fischer, Carl Staelin. **Detecting and exploiting near-sortedness for efficient relational query evaluation.** ICDT 2011: 256-267