



## CS561 Spring 2026 - Research Project

**Title:** *Back to Front Compaction in LSM trees*

**Background:** LSM Trees store data in immutable Sorted String Tables (SSTs) [1]. Traditionally, the SSTs are sorted in ascending order. During compaction the input runs are processed head to tail, and the ascending order is also maintained in the output run. Due to the properties of the files, trimming the data from the head of the file is expensive. Therefore, even after processing the blocks from the head of the input file, they cannot be discarded on the fly. This leads to space amplification.

However, since trimming the tail of the file is cheaper, the input runs can be processed in descending order and the processed blocks can be discarded. This way, as input files are processed, they shrink, and the output files grow in the opposite direction. Keeping SSTs in ascending order is just a convention and not a strict requirement. Range scans, binary searches and merges can work with descending direction as well. Therefore, tail to head compaction will work, as long as the input runs have a consistent ordering regardless of the direction.

**Objective:** This project aims to explore how tail to head compaction can be used to reduce the space amplification by allowing trimming the input files on the fly.

**Steps:** This project would require the students to familiarize themselves with traditional LSM compactions. Here we provide few steps to get started with the project:

1. Measure the space amplification caused by the conventional head-to-tail compaction strategy
2. Design the new tail-to-head compaction strategy and analyze its impact on space amplification

**Responsible mentor:** *Teona Bagashvili*

### References:

[1] Luo, Chen, and Michael J. Carey. "LSM-based storage techniques: a survey." *The VLDB Journal* 29, no. 1 (2020): 393-418. <https://arxiv.org/abs/1812.07527>