Course Mini-Project #4: Implementation of Dictionary Encoder

Due date: March 30

1. Introduction

The objective of this mini-project is to implement a dictionary encoder. As discussed in class, dictionary encoding is being widely used in real-world data analytics systems to compress data with relatively low cardinality. In essence, a dictionary encoder scans the to-be-compressed data to build the dictionary that consists of all the unique data items, and replaces each data item with its location ID in the dictionary. As discussed in class, to accelerate dictionary look-up, you may want to use indexing data structures such as hash-table or B-tree. Meanwhile, a dictionary encoder must support both "extract" that converts an encoded data (i.e., the location ID) to the original data item, and "locate" that converts an original data item to its encoded ID.

2. Requirement

Your implementation should support the following operations:

(1) Encoding: given a raw data file consisting of raw column data, carry out dictionary encoding and generate an encoded column file consisting of both dictionary and encoded data column
(2) Query: enable users to query an existing encoded column file. In particular, your implementation should allow users to check whether one data item exists in the column, if it exists, count the total number of its occurrence in the column

Your implementation should use SIMD instructions whenever possible. You may also consider using multi-threading to speed up the dictionary encoding process (not required). In addition to the source code, your Github site should contain

(1) Readme that clearly explains the structure/usage of your code
(2) Experimental results that show the performance of your dictionary encoder (both encoding performance and query performance)
(3) Analysis and conclusion