

# CS 293 Lab Assignment 6

September 9, 2022

In this lab, we are going to build tries and implement the Knuth-Morris-Pratt algorithm for pattern matching in strings.

You are provided a fairly comprehensive harness code that implements a journey planner. We will add functionality to this to build the overall journey planner we aim to generate by the end of the semester.

For the current assignment, you should be concerned only with files `Trie.h`, `Trie.cpp` and `kmp.cpp`. Please of course feel free to peep into the other files if that helps.

We want to use tries to do completion of lookup of station names. A station name can have multiple space separated parts like “Mumbai Central”, “Delhi Cantonment”, “Bangalore Central” etc.. The trie should allow you to search for words in a station name and present a list of possible completions that the user can select. For example, on searching for “Central”, the trie should allow you to see the completions “Mumbai Central” and “Bangalore Central”, from which the user can choose one option. Therefore, given a (multi-word) station name, you have to insert each word into the trie, and then at the leaf you have to store an index/pointer to the full name of the station stored as an set/array/list of strings containing the full names.

We also want users to be able to enter reviews of journeys (identified by journey code, source station and destination station). These will be stored in a two dimensional array, where the row and column indices are obtained by hashing the source and destination (full) station names respectively. Each entry in the two dimensional array is a list of (journey code, list of reviews). Thus, we will store reviews as a two level list. At the first level, it is a list of journey codes. For each journey code, we will have a list of reviews for that journey code.

A user should also be able to find reviews of a journey (identified by journey code, source station and destination station). When finding reviews, the user has the option of providing a keyword to search for in the reviews, like “good”, “convenient”, “terrible”, “avoidable” etc. Once the journey details (journey code, source station, destination station) is given and a keyword is given, you should use KMP algorithm to find all reviews for that journey in which there is at least one occurrence of the keyword. After that, the review should be printed on the screen.

While all of this sounds complicated, the harness code already does most of the dirty work for you. You should simply implement the trie functions and KMP algorithm and the rest should happen on its own.

For compilation and submission instructions, please check Moodle.