

Programming Assignment #7

Hashing

1 Problem Description

Hashing is used to index and retrieve items in a large amount of data because it is faster to find the item using the shorter hashed key than to find it using the original value. It is also used in many encryption algorithms. A hash function is any function that can be used to map data of arbitrary size to fixed-size values. With proper implementation of both hash function and hash table, the insert, delete, search operations can be performed in constant time, which is much better than $O(\lg n)$ with tree data structures.

In this programming assignment, you will design your own hash function and hash table. Given a set of strings and search targets, you are asked to return all strings with exactly the same characters as those in the target, while ignoring the sequence of the characters.

2 Input Description

For each case, there are two input files, including “strings.in” and “targets.in”.

“strings.in” contains a sequence of at most 20M strings, and each string consists of as large as 20 random characters from “a” to “z”, all in lower case. You will need to construct a hash table after inputting “strings.in”.

“targets.in” contains a set of at most 10K search targets. Each search target is also a string containing at most 20 random characters from “a” to “z”.

“strings.in”	“target.in”
abc	dbe
bde	bca
ec	ac
bac	ce
abb	dbe
edb	abb
a	abc
ce	
bed	
ace	
abc	

3 Output Description

You will need to generate an output file “strings.out” containing the strings in “strings.in” which exactly match the search targets in “targets.in”. The matching criteria is that all characters in an input string are the same as all characters in a search target without the consideration of the order of characters in the strings.

For each search target, the matched strings should be printed in the same line of the output file with a space character in between. The matched strings should also follow the input sequence. If none of the strings is matched, you must output “0”, instead.

“strings.out”
bde edb bed
abc bac abc
0
ec ce
bde edb bed
abb
abc bac abc

4 Command-line Parameter

In order to correctly test your program, you are asked to add the following command-line parameters to your program.

[executable file name] [input file name] [target file name] [output file name]

(e.g., StudentID.exe sample_input.in sample_target.in sample.out)

5 Submission Information

Your program must be written in the C/C++ language, and can be compiled on the Linux platform. The source files of your program must be named with “[your student ID].h” and “[your student ID].cpp”. The executable file name of your program must be “[your student ID].exe”. To submit your program, please archive both executable and source files of your program into a single zip file, named “[your student ID].zip”, and upload to E3.

6 Due Date

The zip file must be submitted through E3 before 23:59, January 12 , 2022.

7 Grading Policy

The programming assignment will be graded based on the following rules:

- Pass sample input with compilable source code (50%)

- Pass five hidden test cases (50%); For each case, you receive the full score with the runtime less than 2 minutes, 80% for 2–5 minutes, 50% for 5–10 minutes, and 0% for longer than 10 minutes.

The submitted source codes, which are copied from or copied by others, will NOT be graded. There will be 25% penalty per day for late submission.