CS202, Spring 2024

Homework 3 - Hashing

Due: 23:55, 30/04/2024

Before you start your homework please read the following instructions carefully:

FAILURE TO FULFIL ANY OF THE FOLLOWING REQUIREMENTS WILL RESULT IN A GRADE SCORE OF 0 (zero) WITHOUT ANY CHANCE OF REDEMPTION.

- See the course page for any late submission policies and Honor Code for Assignments.
- Upload your solutions in a single ZIP archive using the Moodle submission form. Name the file as studentID_name_surname_hw3.zip.
- Your ZIP archive should contain **only** the following files:
  - **studentID_name_surname_hw3.pdf**, the file containing the answers to Questions 1, 2 and 4.
  - In this assignment, you must have separate interface and implementation files (i.e., separate .h
  - and .cpp files) for your class. Your class name MUST BE **HashTable** and your file names MUST BE
  - **HashTable.h** and **HashTable.cpp**. You have to implement **Subtask1.cpp**, **Subtask2.cpp** and **Subtask3.cpp** to solve and test subtasks.Note that you may write additional class(es) in your solution.
  - Your .cpp, .h , .pdf files, and **Makefile** , which produces executables for each subtask (No Makefile results in 50% deduction in points).
  - Do not forget to put your name, student id, and section number in all of these files. Comment your implementation well. Add a header (see below) to the beginning of each file:

    ```
    /**
     * Title: Hash Table
     * Author : Name & Surname
     * ID: 12345678
     * Section : 1
     * Homework : 3
     * Description : description of your code
     */
    ```
  - Do not put any unnecessary files such as the auxiliary files generated from your preferred IDE.
  - Please do not use **Turkish** letters in your file and folder names.
- Your code must compile.
- Your code must be complete.
- You ARE NOT ALLOWED to use any data structure or algorithm related function from the C++ standard template library (STL) or any other external libraries.
- We will test your code using the **google test library**. Therefore, the output message for each operation in Question 2 and 3 MUST match the format shown in the output of the example code.
- Your code must run on the **dijkstra.cs.bilkent.edu.tr** server.
- For any question related to the homework contact your TA: *ziya.ozgul@bilkent.edu.tr*

Question 1 - 5 points

Assume that we have a hash table with size 13 (index range is 0..12) and we use the mod operator as a hash function to map a given numeric key into this hash table. Draw the hash tables after the insertion of

keys 15 13 9 12 28 11 25 24 in the given order for each of the following methods:

- open addressing with linear probing
- open addressing with quadratic probing
- separate chaining

You MUST have a handwritten answer for this question.

Question 2 - 8 points

Assume that you have n integers and you have a hash table with hash function, f(x) = x % tableSize. However, you don't know the value of tableSize. You must implement a function minTableSize(int *array, int n) which determines the minimum size of hash table such that there will be no collision. Also, the size of the hash table must be **prime.** You are not allowed to use any auxiliary space for this question and not allowed to use mod (%) operation.

```
Sample Input

5
6
7
8
9
10

Sample Output

5
```

Explanation : 6 % 5 = 1, 7 % 5 = 2, 8 % 5 = 3, 9 % 5 = 4, 10 % 5 = 0. These are unique.

You MUST have a handwritten answer for this question.

Question 3 - 72 points

Subtask 1

You are given n unique strings. You want to know the number of pairs (i,j) such that the ith string is either prefix or suffix of the jth string and $i \neq j$. To make it clear, an example is given below.

```
Sample Input( strings.txt )

10
ab
abc
abdce
bc
bcda
c
cab
bca
def
ef

Sample Output( pairs.txt)

10
```

Explanation:

1 -> “ab” is prefix of “abc”   2 -> “ab” is prefix of “abcde”
3 -> “ab” is suffix of “cab”   4 -> “bc” is suffix of “abc”
5 -> “bc” is prefix of “bcda”   6 -> “bc” is prefix of “bca”
7 -> “c” is suffix of “abc”   8 -> “c” is suffix of “bc”
9 -> “c” is prefix of “cab”   10 -> ef is suffix of “def”

Important:

If one is both prefix and suffix of another, you need to count it twice.
For example, “a” is both prefix and suffix of “aba”. So you count it twice.

- This homework MUST be done using hashing.
- If you try to solve this question using data structures other than hash tables you will get 0 from this part and FZ from this course.
- You can implement hash tables using the implementations in lecture slides.
- Your solution must be efficient. In other words, we don't want you to try all possible pairs. We want you to determine the possible pairs using hash tables.
- You MUST implement your hash table using separate chaining.
- To get a full credit your code must work correct and work in O(total_number_of_characters).

The name of the input file will be provided as a command line argument to your program. Thus, your program should run using two command line arguments. Thus, the application interface is simple and given as follows:

```
username@dijkstra:~>./subtask1 <input_filename> <output_filename>
```

Assuming that you have an executable called “subtask1”, this command calls the executable with one command line argument, the name of the file from which your program reads the number of strings and strings.

Note that outputfile name is optional.

Subtask 2

You are given one text , length of n, and m unique patterns. You want occurrences of each pattern in text. You are given text in the first line. Number of patterns are given in the second line. Patterns are given afterwards. It is given that difference between the length of patterns can be at most 5.

```
Sample Input( patterns.txt )
abcbccdaaabcdbccd
4
bc
ab
cd
bcc
Sample Output( occurrences.txt)

4
2
3
2
```

Explanation:

First pattern, “bc”, occurs at 2,4, 11 and 14 indices(one based indexing).
Second pattern, “ab”, occurs at 1 and 10.
Third pattern, “cd”, occurs at 6,12 and 16.
Forth pattern, “bcc”, occurs at 4 and 14.

- This subtask MUST be done using hashing.
- You don’t have to use separate chaining like previous subtask.
- To get a full credit your code must work correct and work in O((n + m) * log m + total_number_of_characters).
- If you try to solve this question using data structures other than hash tables you will get 0 from this part and FZ from this course.
-The name of the input file will be provided as a command line argument to your program. Thus, your program should run using two command line arguments. Thus, the application interface is simple and given as follows:

```
username@dijkstra:~>./subtask2 <input_filename> <output_filename>
```

Assuming that you have an executable called “subtask2”, this command calls the executable with one command line argument, the name of the file from which your program reads the number of strings and strings.

Note that outputfile name is optional.

Subtask 3

You are given n unique strings. You can make two operations on strings. The first one is reverse operation. For example, the string “abcd” becomes “dcba” after reverse operation. Second one is the shift operation. For example, the string “abcd” becomes “bcda” after shift. “bcda” becomes “cdab”. However, there is a restriction. You have to complete all reverse operations before you start shift operations. Doing these operations, you can make strings equal. Your aim is to find the minimal subset of given strings. Minimal subset is defined as a subset of given strings with minimum number of elements. In other words, when you reach a minimal subset you cannot minimize the

strings further. For example, you have strings “abcd” and “bcda”. When you apply shift one first one you only have “bcda” and this is a minimal subset. Note that there can be different minimal subsets but the size of the minimal subset is unique. However, you are required to minimize the number of reverse operations when you make the given set minimal. On the other hand, there is no limitation to shift operations.
You have a number of strings in the first line and you have strings after that.
You are expected to print the number of reverse operations(first line of output) and size of minimal subset(second line).

```
Sample Input 1( strings.txt )
2
abcde
edcba
Sample Output( output.txt)
1
1
```

Explanation:

The size of the minimal subset is 1. Minimal subset can be either “abcde” or “edcba” and its size is 1. You need to have one reverse operation on “abcde” to make it equal to “edcba” or vice versa.

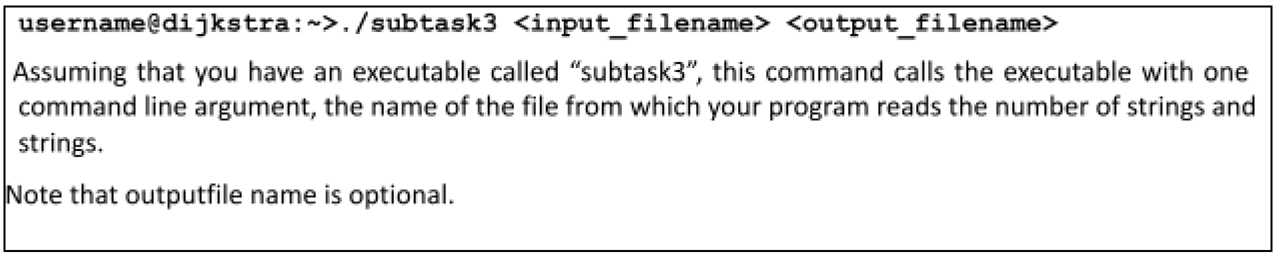
line).

```
Sample Input 2( strings.txt )
4
baaa
aaab
bcd
dcb
Sample Output( output.txt)
1
2
```

Explanation:

The size of the minimal subset is 2. Minimal subset can be {“baaa”, “bcd”} and its size is 2. You need to have one reverse operation on “bcd” to make it “dcb” or vice versa. “baaa” is the reverse of “aaab” but you don’t need reverse operation. When you apply shift on “baaa”, you obtain “aaab”.

- This subtask MUST be done using hashing.
- If you try to solve this question using data structures other than hash tables you will get 0 from this part and FZ from this course.
- To get a full credit your code must work correctly and work in O(total_number_of_characters * log n) (or better).
-The name of the input file will be provided as a command line argument to your program. Thus, your program should run using two command line arguments. Thus, the application interface is simple and given as follows:



```
username@dijkstra:~>./subtask3 <input_filename> <output_filename>
```

Assuming that you have an executable called “subtask3”, this command calls the executable with one command line argument, the name of the file from which your program reads the number of strings and strings.

Note that outputfile name is optional.

Important: The number of strings <= 100000 and total number of characters in strings <= 10000000 in each test case for each subtask. You need to make your calculations based on this. You need to determine your modulus(hash table sizes) based on this to handle collisions.

Important Note for Outputs

You can add the output file name as an argument to print the outputs of your code.
When I grade your homeworks, I could also give output file name as a parameter to your program.
Therefore, please add this script to your implementation for each subtask.

```
!!!!!
if ( argc > 2 ){
    freopen(argv[2], “w”, stdout);
}
!!!!!
```

Question 4 - 15 points

In this question, you will analyze your solutions for Question 3 for each subtask. You must discuss the question, discuss about the worst case scenarios. You also need to discuss your solution. You need to talk about your hash table and features of hash table.
You are expected to mention hash functions. Also, you need to talk about how you handle collisions. What have you done for the collision scenarios? You MUST NOT write this part in only 2-3 sentences. It should be detailed enough to get a full credit.