CSE 4304-Data Structures Lab. Winter 23-24

Batch: CSE 22

Date: November 13, 2024

Target Group: All

T opic: Hashing

Instructions:

- Regardless of how you finish the lab tasks, you must submit the solutions in Google Classroom. In case I forget to upload the tasks there, CR should contact me. The deadline will always be 11:59 PM on the day the lab took place.
- Task naming format: fullID_T01L01_2A.c/cpp
- If you find any issues in the problem description/test cases, comment in the Google Classroom.
- If you find any tricky test cases that I didn't include but that others might forget to handle, please comment! I'll be happy to add them.
- Use appropriate comments in your code. This will help you to recall the solution in the future easily.
- Obtained marks will vary based on the efficiency of the solution.
- Do not use <bits/stdc++.h> library.
- Modified sections will be marked with BLUE color.
- You are allowed to use the STL stack unless it's specifically mentioned to use manual functions.

Group	**Tasks**
2A	1 2 3
1B	1 2 3
1A	1 4 5 BONUS: 6
2B	1 4 5 6
Assignments	2A/1B: 4 5 6 7 8 9 1A/2B: 2 3 6 7 8 9

Task-1

Suppose you are implementing a Hash Table, but your hash function can’t guarantee to provide a unique index to each key. Hence you need to adopt some of the collision resolution techniques f(i).

Implement the following Collision handling techniques:

1. Linear Probing: f(i) = i
2. Quadratic Probing: f(i)=i^2
3. Double Hashing: f(i)=i*hash2(x); hash2(x) = R - (x % R) with R=7

The first input line should be (choice, N, Q), where ‘choice’ can be 1/2/3 corresponding to linear/quadratic/double hashing. N represents the size of the HashTable. Q represents the number of queries.
Then, there will be Q numbers given as input.

- Main Hash function: Hash(x) = (x % TableSize)
- After each insertion, print the Load Factor (L.F.) representing the ratio between #number of inserted items and tableSize.

Sample Input	**Sample Output**

1 10 8
35
45
73
36
5
24
13
99
	Inserted : Index-5 (L.F=.1)
	Collision: Index-5
	Inserted : Index-6 (L.F=.2)
	Inserted : Index-3 (L.F=.3)
	Collision: Index-6
	Inserted : Index-7 (L.F=.4)
	Collision: Index-5
	Collision: Index-6
	Collision: Index-7
	Inserted : Index-8 (L.F=.5)
	Inserted : Index-4 (L.F=.6)
	Collision: Index-3
	Collision: Index-4
	Collision: Index-5
	Collision: Index-6
	Collision: Index-7
	Collision: Index-8
	Input Abandoned
	Inserted : Index-9 (L.F=.7)

2 8 7
67
15
86
63
47
	Inserted : Index-3 (L.F = 0.125)
	Inserted : Index-7 (L.F = 0.25)
	Inserted : Index-6 (L.F = 0.375)
	Collision: Index-7
	Inserted : Index-0 (L.F = 0.5)
	Collision: Index-7
	Collision: Index-0
	Collision: Index-3
	Collision: Index-0
	Collision: Index-7
	Collision: Index-0
	Input Abandoned

33	Inserted : Index-1 (L.F = 0.625)
8	Collision: Index-0 Collision: Index-1 Inserted : Index-4 (L.F = 0.75)
3 15 11	
94	Inserted : Index-4 (L.F = 0.0666667)
46	Inserted : Index-1 (L.F = 0.133333)
61	Collision: Index-1 Inserted : Index-3 (L.F = 0.2)
29	Inserted : Index-14 (L.F = 0.266667)
85	Inserted : Index-10 (L.F = 0.333333)
77	Inserted : Index-2 (L.F = 0.4)
46	Collision: Index-1 Collision: Index-4 Inserted : Index-7 (L.F = 0.466667)
63	Collision: Index-3 Collision: Index-10 Collision: Index-2 Inserted : Index-9 (L.F = 0.533333)
67	Collision: Index-7 Collision: Index-10 Inserted : Index-13 (L.F = 0.6)
93	Collision: Index-3 Inserted : Index-8 (L.F = 0.666667)
61	Collision: Index-1 Collision: Index-3 Inserted : Index-5 (L.F = 0.733333)

Note:

- If an item cant be inserted within six attempts, abandon that item.

(Please test your program for different TableSize and different sets of numbers)

Task 2

Given a collection of integers and a number ‘target’, find the pairs of integers whose summation is equal to ‘target’. The elements of the collection may not be unique.

The first line provides the collection of integers where -1 denotes the end of the input. The following line will contain the target value.

Output:

- Print the pairs whose summation equals ‘target’.
- If none of the pairs adds up to ‘target’, print ‘No pairs found’.

Sample Input	**Sample Output**
2 5 4 12 9 1 3 17 11 8 -1 13	(8,5), (12,1), (9,4), (11,2)
2 2 2 -1 4	(2,2)
2 5 4 2 0 1 3 -1 4	(2,2), (3,1), (4,0)
2 5 4 2 0 2 7 -1 6	(4,2)
2 5 4 12 9 1 3 17 8 11 8 5 -1 13	(8,5), (12,1), (9,4), (11,2), (5,8)
1 1 1 2 2 2 -1 3	(1,2), (1,2), (1,2)
1 1 1 1 2 2 1 1 1 1 2 -1 3	(1,2), (1,2), (1,2)
4 -2 2 7 9 1 3 1 0 -1 7	(4,3), (7,0), (9,-2)
2 5 4 12 9 1 3 17 11 8 10 -1 100	No pairs found

Note:

- Explore the ‘unordered_map’ library function (https://www.geeksforgeeks.org/unordered_map-in-cpp-stl/)
- The elements/pairs can appear in any order. Hence [(8,5), (12,1), (9,4), (11,2)] and [(5,8), (1,12), (4,9), (2,11)] mean the same. There can be other combinations.
- **Rejected Solution:** Store the collection in an array. For every element, search the (target-current) element from the remaining portion of the array. Complexity O(n^2).
- Provide O(n) solution with Hashmaps

Task 3:
Given a sentence, you have to find the word(s) that occur more than once. Ignore the punctuation marks.

Input	**Output**
data atad structure atad data	data 2 atad 2
I know you know this, but you do not know of unknown trolls, because no known trolls will sew by windows, though they will owe you a hello when they throw a hoe, as it will go low and blow a hole in that window and so it will follow, that it happened awhile ago, as a troll will stand on a knoll and show you how to throw snow tomorrow at a rhino named Joe, who plays piano as he sips a cappucino and sings soprano in an inferno caused by a volcano in Reno with a casino at the bottom. Of the volcano.	volcano 2 a 9 by 2 will 5 at 2 throw 2 in 3 you 4 it 3 the 2 know 3 and 4 trolls 2 they 2 as 3 that 2
I know you know this but you do not know of unknown trolls because no known trolls will sew by windows though they will owe you a hello when they throw a hoe as it will go low and blow a hole in that window and so it will follow that it happened awhile ago as a troll will stand on a knoll and show you how to throw snow tomorrow at a rhino named Joe who plays piano as he sips a cappucino and sings soprano in an inferno caused by a volcano in Reno with a casino at the bottom. Of the volcano	volcano 2 a 9 by 2 will 5 at 2 throw 2 in 3 you 4 it 3 the 2 know 3 and 4 trolls 2 they 2 as 3 that 2
This refers to an exam where James had written ‘had had’ where John had written just ‘ had’. The examiner had approved James’ version.	James 2 where 2 had 6 written 2
This refers to an exam where James had written had had where John had written just had The examiner had approved James version	James 2 where 2 had 6 written 2

Note:

- Show the words in any order.
- May use **getline**(cin, sentence) to take the input.

Task 4

Given an array of points where points[i] = [xi, yi] represents a point on the X-Y plane, return the maximum number of points that lie on the same straight line along with the equation of the straight line.

Input: N coordinates

Output : The line through which maximum number of points pass through, and the number of points lying on that line.

Sample Input	Sample Output
10 0 −4 0 4 1 −2 1 6 3 2 2 1 1 1 0 -1 4 4 0 0	y = 2.00x - 4.00 4

3
1 1
2 2
3 3
	y = 1.00x
	3

6
1 1
3 2
5 3
4 1
2 3
1 4
	Y = -1.00x + 5.00
	4

- The format of the equation does not matter as long as it is correct
- The complexity of your solution must be in O(n^2)
- You are allowed to use unordered_map

Task 5

International Morse Code defines a standard encoding where each letter is mapped to a series of dots and dashes, as follows:

- 'a' maps to ".-"
- 'b' maps to "-..."
- 'c' maps to "-.-." and so on.

For convenience, the full table for the 26 letters of the English alphabet is given below:

{".-","-...","-.-.","-..",".","..-.","--.","....","..",".---","-.-",".-..","--","-.","---",".--.","--.-",".-.","...","-","..-","...-",".--","-..-","-.--","--.."}

Given an array of strings words where each word can be written as a concatenation of the Morse code of each letter.

For example, "cab" can be written as "-.-..--...", which is the concatenation of "-.-.", ".-", and "-...". We will call such a concatenation the transformation of a word.
Return the number of different transformations along with the transformations among all words we have.

Sample Input	Sample Output
4 gin zen gig msg	2 --...-. --...--.
6 gin zen gig msg cab dog	4 --...-. --...--. -.-..--... -..-----.
1 a	1 .-

Task 6

One of the most popular and students' favorite teachers, SA, is trying to build a game called MAC-MAN. It’s a game where a character runs through a maze and is chased by enemies, aiming to collect special items scattered across the map (sounds familiar, right?). The goal is to score the highest points by collecting these elements while avoiding being caught by the enemies.

However, SA is currently swamped with checking the Data Structures scripts of 120 students, and he’s starting to feel the pressure. Since he doesn’t have enough time to manually design each maze, he had an idea. He decided to assign his students the task of designing different mazes, filled with special characters (like @, #, $, %, *, etc.), which would help him save time.

Now, being the clever and witty teacher that he is, SA anticipated that some students might try to outsmart him by submitting someone else’s work. He knows how some students tend to copy each other's assignments. But since he’s already overwhelmed with work, SA cleverly delegated the task of finding out who copied from whom to you, the budget friendly Sherlock.

You are given three values at the beginning:

T: The number of submitted works (maps).
M and N: The dimensions of the maze (M x N grid).

Then, you will be provided with T cases, each containing a **student's name** and the **maze** that they have designed. Your task is to identify students whose maze designs are identical.

For each case, compare the maze designs and print the names of the students who have submitted identical maps.

Sample test Case
Sample Output

Hints

1. Use maps/hashing,
2. Do not compare mazes with each other, your solution must be in O(n^2)

Task 7:

The DNA sequence is composed of a series of nucleotides abbreviated as 'A', 'C', 'G', and 'T'. For example, "ACGAATTCCG" is a DNA sequence. When studying DNA, it is useful to identify repeated sequences within the DNA.

Given a string ‘s’ that represents a DNA sequence, return all the 10-letter-long sequences that occur more than once in a DNA molecule. You may return the answer in any order.

Constraints:

- 1 <= s.length <= 10^5
- s[i] is either 'A', 'C', 'G', or 'T'.

Sample Input	**Sample Output**
AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCC, CCCCCAAAAA
AAAAAAAAAAAAAAAA	AAAAAAAAAA

Note:

- Use the Rabin-Karp string-matching algorithm to calculate the hash value. Don’t use unordered maps for this task.
- Implement the **‘Rabin-Karp String Matching Algorithm’** using the concept of the **rolling-hash function**. Test your program for different test cases. Make sure you understand how this algorithm is improving the traditional approach

Task 8:

Given a sentence and a word, your task is to print the words that consist of the same **unique character set.**

The first line of input will contain the sentence, and the second line will contain the word. Print the word(s) that consist of the same unique character set (length doesn't matter)

Input	**Output**
You may know the answer but it is not yam or maaayaaay or yammy may	may yam maayaaay yammy
student will act like students and it is studentish dont write studnet by mistake student	student students studnet
abcd abd acd bad baad baacd aabbccdd accc aadd abbe aaag aabbcd	abcd baacd aabbccdd
aceg aceegg ggccaaaee bcdg hbae aceg	aceg aceegg ggccaaaee

Note:

- Write a hash function which will check the similarity between two strings utilizing the unique characters. Just adding the ASCII of the unique characters won't work as 'ad' and 'bc' gives the same result. Need a better hash value.
- Explanation for input-1: The unique character set in 'may' is 'm,a,y'. As 'maaayaaay' also consists of the same set of unique characters 'm-a-y' it is a part of the output.
- All the unique characters of the word must be used. Hence, the word 'aabbcd'-'abd', 'acd' etc is not a match as one of the unique characters are missing.
- The hash value of two words having the same unique character set should be equal.
- May use **getline**(cin, sentence) to take the input.

Task 9 – Babelfish

Problem Statement

You have just moved from Waterloo to a big city. The people here speak an incomprehensible dialect of a foreign language. Fortunately, you have a dictionary to help you understand them.

Input

Input consists of several dictionary entries, followed by a blank line, followed by a message consisting of many words. Each dictionary entry is a line containing an English word, followed by a space and a foreign language word. No foreign word appears more than once in the dictionary. The message is a sequence of words in the foreign language, one word on each line.

Output

Output is the message translated to English, one word per line. Foreign words not in the dictionary should be translated as **‘eh’**.

Sample Test Case(s)

Input	**Output**
dog ogday cat atcay pig igpay froot ootfray loops oopslay atcay ittenkay oopslay	cat eh loops