National Institute of Technology, Calicut

Department of Computer Science and Engineering

CS2094D – Data Structures Lab

Assignment-2 (Advanced Batch)

Policies for Submission and Evaluation

You must submit your assignment in the moodle (Eduserver) course page, on or before the submission deadline. Also, ensure that your programs in the assignment must compile and execute without errors in Athena server. During evaluation your uploaded programs will be checked in Athena server only. Failure to execute programs in the assignment without compilation errors may lead to zero marks for that program.

Your submission will also be tested for plagiarism, by automated tools. In case your code fails to

pass the test, you will be straightaway awarded zero marks for this assignment and considered by the examiner for awarding F grade in the course. Detection of ANY malpractice regarding the lab course will also lead to awarding an F grade.

Naming Conventions for Submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>.zip (For example: ASSG4_BxxyyyyCS_LAXMAN.zip). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive. The source codes must be named as ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>_<PROGRAM-NUMBER>.<extension> (For example: ASSG4_BxxyyyyCS_LAXMAN_1.c). If there is a part a and a part b or a particular question, then name the source files for each part separately as inASSG4_BxxyyyyCS_LAXMAN_1b.c.

If you do not conform to the above naming conventions, your submission might not be recognized by some automated tools, and hence will lead to a score of 0 for the

submission. So, make sure that you follow the naming conventions.Standard of Conduct Violations of academic integrity will be severely penalized.

Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course.

The department policy on academic integrity can be found at:
http://cse.nitc.ac.in/sites/default/files/Academic-Integrity.pdf .

Assignment Questions

Q. 1) Write a C program to implement hash table data structure to store student's information with roll number as key. Input should be read from the file ***input.txt*** and output should be written to the file ***output.txt*****.** Your program should contain the following functions:-

- **hashTable(int m**)- create a hash table of size m

-**insert(int k)**- insert element into hash table having key value as k

-**search(int k)**- find whether element with key 'k' is present in hash table or not

-**delete(int k)**- delete the element with key 'k'

Input Format:

The first line contains a character from { 'a', 'b', 'c', 'd'} denoting

1. a- Collision resolution by Linear probing with hash function

   $h(k,i) = (h_1(k) + i) \bmod m$ where $h_1(k) = k \bmod m$

2 b - Collision resolution by Quadratic probing with hash function

$h(k,i) = (h_1(k) + c_1 i + c_2 i^2) \bmod m$ where $h_1(k) = k \bmod m$) c1 and c2 are positive auxiliary constants, i = 0,1,... m-1

3 c - Collision resolution by Double Hashing with hash functions

$h(k,i) = (h_1(k) + i * h_2(k)) \bmod m$ where,

$h_1(k) = k \bmod m$,

$h_2(k) = R - (k \bmod R)$ { R = Prime number just smaller than size of table }

4 d - Collision resolution by Chaining with hash function

$h(k) = k \bmod m$ Where,

k = Key, m = size of hash table

Next line contains an integer, m, denoting the size of hash table.

In case of quadratic probing only (option b) Next line line contains the constants c1 and c2 separated by space

Next lines contains a character from { ‘i’, ‘s’, ‘d’, ‘p’, ‘t’ } followed by zero or one integer.

● i x - insert the element with key x into hash table

● s x - search the element with key x in hash table. Print 1 if present otherwise print -1

● d x - delete the element with key x from hash table.

● p - print the hash table in “index (key values)” pattern. (See sample output for explanation)

● t - terminate the program

(Note : ● In case of Linear probing, quadratic probing and Double hashing, total elements (n) to be nserted into hash table will be lesser than or equal to size of the hash table (m) i.e. n<=m

● deletion operation will always be a valid operation

● While printing the hash, multiple key values must be separated by a single white space.)

Output File Format:

The output (if any) of each command should be printed on a separate line

Sample Input

a

7

i 76

i 93

i 40

i 47

i 10

i 55

p

s 35

s 47

d 47

s 55

t

Sample Output

0 (47)

1 (55)

2 (93)

3 (10)

4 ()

5 (40)

6 (76)

-1

1

1

Sample Input

b

7

0 1

i 76

i 40

i 48

i 5

s 5

i 55

p

s 62

d 55

Sample Output

1

0 (47)

1 ()

2 (5)

3 (55)

4 ()

5 (40)

6 (76)

-1

1

-1

Sample Input

c

7

i 76

i 93

i 40

i 47

i 10

i 55

p

d 40

s 47

s 76

s 40

t

Sample Output

0 ()

1 (47)

2 (93)

3 (10)

4 (55)

5 (40)

6 (76)

1

1

-1

Sample Input:

d

5

i 25

i 11

i 10

i 21

i 15

p

s 26

s 15

d 10

s 10

t

Sample Output

0 (25 10 15)

1 (11 21)

2 ()

3 ()

4 ()

-1

1

-1

Q. 2) Write a menu driven program to implement the Rabin-Karp Algorithm for pattern searching.

Your program must implement the following functions:

main() - repeatedly reads a character 's' or 'p' from the file **input.txt** and calls the given functions appropriately until character 'e' is entered.

text()- reads the text from the file **input.txt**

pattern()- reads the pattern from the terminal.

print()- function to print the starting index using Rabin – Karp Algorithm for String Matching, if pattern is found or print -1 in case of no match.

We can compute the hash function as follows:Here d is 10 as text is made of numbers, d must be 256 for your program (Refer: Introduction to Algorithms , Cormen et al. 3rd Edition , Chapter 32)

Lets assume, $h_0 = k$

$$h_1 = d\left(k - p[1].d^{m-1}\right) + p[m+1]$$

Suppose, we have given a text $t = [3, 1, 4, 1, 5, 2]$ and m = 5, q = 13;

$t_0 = 31415$

So $t_1 = 10(31415 - 10^{5-1}.t[1]) + t[5+1]$

$= 10(31415 - 10^4.3) + 2$

$= 10(1415) + 2 = 14152$

Here p and substring t_i may be too large to work with conveniently. The simple solution is, we can compute p and the t_i modulo a suitable modulus q.

So for each i,

$$h_{i+1} = \left(d\left(h_i - t[i+1].d^{m-1}\right) + t[m+i+1]\right) \bmod q$$

The modulus q is typically chosen as a prime such that d.q fits within one computer word.

Algorithm

Compute h_p (for pattern p)
Compute h_t (for the first substring of t with m length)
For $i = 1$ to $n - m$
  If $h_p = h_t$
    Match $t[i \ldots . i + m]$ with p, if matched return 1
  Else
    $h_t = \left(d\left(h_t - t[i+1].d^{m-1}\right) + t[m+i+1]\right) \bmod q$
  End

Suppose, t= 23590231415267399 21 and $p = 31415$,

Now, $h_p = 7$ $(31415 = 7 \pmod{13})$

substring beginning at position 7 = valid match

Input Format :

a) The input consists of multiple lines. Each line contains a character from {'t', 'p', 'pr', 'e' } followed by text.

b) Character 't': character ‘t’ specifies the text.

c) Character 'p': character ‘p’ specifies the pattern.

d) Character 'pr': character ‘pr’ specifies the print function.

e) Character 'e': character ‘e’ specifies the end of the program.

Sample Input:

t “THIS IS A TEST TEXT”

p “TEST”

pr

t “AABAACAADAABAABA”

p “AABA”

pr

e

Sample Output:

10
0 9 12

Q. 3) Write a menu driven program to implement a C program to find all symmetric pairs in an array using hash table , If an array of pairs is given as input. Input should be read from **input.txt** and output(if any) should be written to **output.txt**

Your program must implement the following functions:

main() - repeatedly reads a character 's' or 'p’ from the file **input.txt** and calls the given functions appropriately until character 'e' is entered.

store()- read the pairs of an array and store the pairs.

print()- function to print the symmetric pairs.

Input Format

In this program you have to choose suitable hash function for storing the pairs.

a) The input consists of multiple lines. Each line contains a character from {'s', 'p’, 'e' } followed by zero or a pair of integers.

b) Character 's': character ‘s’ specifies to store the pairs of an array.

c) Character 'p': prints the symmetric pairs of an array.

d) Character 'e': character ‘e’ specifies the end of the program.

Sample Input:

s (11, 20)
s (30, 40)
s (5, 10)
s (40, 30)

s (10, 5)

p

e

Sample Output:

(30, 40)

(5, 10)