

The Chinese University of Hong Kong
Department of Systems Engineering and Engineering
Management

CSCI-2100-B Data Structures

2013-14 Assignment 3

Hashing

We have discussed hashing in lecture-note `ch8.pptx`. In this assignment, you are requested to write a C program by yourself, called `hash` to understand hash functions and open addressing when handling overflows.¹

The coding for hashing is simple. The important thing in this assignment is to understand how the following parameters will affect the searching: the number of buckets b , the number of slots s , the selection of a hash function $h()$ (division and mid-square), and open addressing approach to handle overflows (linear probing).

You will be given a set of non-negative keys to insert into a hash table ht of b buckets where the bucket numbers are in the range of $[0, b - 1]$ and each bucket has s slots. The number of non-negative **distinct** keys to be inserted is less than or equal to $b \times s$.

- Inserting a key k into the hash table ht , using a hash function $h(k)$ (division or mid-square). Suppose the hash function $h(k)$ hashes k into a bucket j for $0 \leq j < b$. If there is an empty slot in the bucket j , then insert k into the empty slot in the bucket j . If all slots in the bucket j are fully occupied, use the linear probing to find a slot to insert it into another bucket repeatedly. It is important to note that if there is an empty slot in the hash table, it must be able to insert k into the hash table ht using linear probing.

¹Departmental Guideline for Plagiarism (Department of Systems Engineering and Engineering Management): If a student is found plagiarizing, his/her case will be reported to the Department Examination Panel. If the case is proven after deliberation, the student will automatically fail the course in which he/she committed plagiarism. The definition of plagiarism includes copying of the whole or parts of written assignments, programming exercises, reports, quiz papers, mid-term examinations and final examinations. The penalty will apply to both the one who copies the work and the one whose work is being copied, unless the latter can prove his/her work has been copied unwittingly. Furthermore, inclusion of others' works or results without citation in assignments and reports is also regarded as plagiarism with similar penalty to the offender. A student caught plagiarizing during tests or examinations will be reported to the Faculty office and appropriate disciplinary authorities for further action, in addition to failing the course.

- Searching a key k in the hash table ht , using the same hash function $h()$ and the same linear probing you used for insertion.
 1. Compute $h(k)$. Suppose the value of $h(k)$ is j , search all slots in the bucket j . If you find it, then you return k and stop searching.
 2. If you cannot find it, check the bucket j for $j = (h(k) + i) \% b$, from $i = 1$ to $b - 1$, one-by-one. In a bucket j , check all slots. If you find it in a slot, then you return k and stop searching. If you find an empty slot in a bucket j during searching, then you can stop searching, because k does not exist in the hash table.

Your program is supposed to be run as follows.

Input: Your program is expected to read from standard input (keyboard) instead of any file, exactly as what you practiced in the previous Lab submissions. The input starts with a number N which denotes the total number of test cases to be followed. For each test case, it begins with four numbers: b which denotes the total number of buckets, s which denotes the the number of slots per bucket, H which denotes a hash function to be used², and n which denotes the total number of keys to be inserted. Here, all the numbers, b , s , and n , are positive numbers, for $n \leq b \times s$. After the three numbers, it follows n non-negative distinct keys to be inserted.

An Input Example: An input example is given below.

```
1
9 1 1 6
10 99 11 4 8 53
```

Here, $N = 1$, $b = 9$, $s = 1$, $H = 1$, and $n = 6$.

Output: In your program, you create an array called Q to keep all the n keys, and you also create a hash table ht to maintain all the n keys. In searching, you pick up every key k you have kept in the array Q , and try to find it in the hash table ht . For searching a key k in the hash table, you know the number of slots you need to access in order to get an answer. (This can be done using a counter in your program.) For each key, your program is expected to output the key k , the home bucket number, and the number of slots you search in order to get this answer in one line. At the end your program is expected to output two numbers to the standard output (screen) in one line: (1) the largest number of slots you need to find a key and (2) smallest number of slots you need to find a key.

Marking: The third programming assignment accounts for 10% of the unit's assessment. Your program will be judged solely on the correctness, i.e., the

²When H is 1, it indicates a division function, $h(k) = k \% b$. When $H > 1$, it indicates a mid-square function, where it takes the middle $r = H$ bits. referring to the slide 8-14 in `ch8.pptx`. Here, if $H > 1$, H will be an even number $0 < H \leq 32$, and a key is a 32-bit integer. In addition, the value of the mid-square should be represented as a 32-bit number (`int` in C).

percentage of test cases that your program can successfully and correctly handle. We will also check your source code. A set of tests will be provided on the course website and eLearning. The set of tests provided is for your testing only, and it does not account for any mark. We will use a different set of tests for marking. We encourage you to submit your program even if you cannot pass the test on online judge, in order for you to receive marks it deserves.

Submission: In every source file you submit, you must include the following at the top. (Replace <Your Full Name> with your full name.)

```
/*  
I, <Your Full Name>, am submitting the assignment for  
an individual project.
```

```
I declare that the assignment here submitted is original except for  
source material explicitly acknowledged, the piece of work, or a part  
of the piece of work has not been submitted for more than one purpose  
(i.e. to satisfy the requirements in two different courses) without  
declaration. I also acknowledge that I am aware of University policy  
and regulations on honesty in academic work, and of the disciplinary  
guidelines and procedures applicable to breaches of such policy and  
regulations, as contained in the University website  
http://www.cuhk.edu.hk/policy/academichonesty/.
```

```
It is also understood that assignments without a properly signed  
declaration by the student concerned will not be graded by the  
teacher(s).
```

```
*/
```

Submit your source file to the Problem No. **1017** in the online judge for this course hosted at <http://sepc498.se.cuhk.edu.hk>. We will use your last submission, as the final submission, for testing.

The due date for the first assignment is 5:00pm, April 18, 2014. The late penalty will be 10% per day. A submission will not be accepted five days after the deadline.

Start working on it as soon as possible!