CSE100 Algorithm Design and Analysis Lab 06, Spring 2022
Deadline: 07-MAR, 11:59pm

Hash Table with Chaining

Description In this assignment you are requested to implement a hash table that handles collisions by chaining. You have to use linked lists, either from the Standard Template Library (STL) (recommended) or by implementing your own. For usage of STL, refer – for instance – to `http://www.cppreference.com/wiki`.

You need to implement the insert, search and delete operations. The keys are integers (C++ int type) and you can assume that all keys are non-negative. The first number in the input will be the size m of the chained hash table. You will use the simple hash function $h(k) = k \mod m$. Then lines will follow starting with 'i', 's', 'd','o', or 'e'. The details are as follows:

- Use the hash function $h(k) = k \mod m$.
- i(key): Insert (key) into the table. For example, "i2" implies "Insert key 2 into the table." For collisions, insert the colliding key at the *beginning* of the linked list. You just need to insert the key and don't have to output anything in this case.
- d(key): delete (key) from the table. For example, d2 implies "Delete key 2 from the table." If there are multiple elements of the same key value, delete the element of the key value that appears the earliest in the list. If the delete was successful, you have to output

  `(key):DELETED;`

  If not (since there was no element with the key value), output

  `(key):DELETE_FAILED;`
- s(key): search (key) in the table. If there is an element with the key value, then you have to output

  `(key):FOUND_ATi,j;`

  where i is the hash table index and j is the linked list index. If there are multiple elements with the same key value, choose the first one appearing in the linked list. If you couldn't find the key, then output

  `(key):NOT_FOUND;`
- o: output the table. Output the entire hash table. Each line should begin with the slot/hash table index followed by key values in the linked list. For example, if $m = 3$ and we inserted 3, 6, and 1 into an empty table in this order, then you should output

  ```
  0:6->3->;
  1:1->;
  2:;
  ```
- e: finish your program.

Example of input and output

The following example shows an execution of the program in interactive mode. See the input files and output files under the testfiles folder for examples where input and output are separated.

```
2
i4
i2
i6
i3
o
0:6->2->4->;
1:3->;
s2
2:FOUND_AT0,1;
s4
4:FOUND_AT0,2;
d5
5:DELETE_FAILED;
d2
2:DELETED;
o
0:6->4->;
1:3->;
e
```

See the lab guidelines for submission/grading, etc., which can be found in Files/Labs.