CSE100 Algorithm Design and Analysis — Lab 09, Fall 2020
Last update: October 19, 2020 — Deadline: Friday, Oct 30, 5:00pm

Hash Table with Chaining

In this assignment you are requested to implement a hash table that handles collisions with 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 here.

You have to implement the insert, search and delete operations. The keys are integers (`C++` int type) and you can assume that all keys k 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 $h(k) = k \mod m$.

The input contains one of the following commands on a line:

- `i key`: Insert `key` into the hash table. For example, "`i 2`" means "Insert key 2 into the hash table". For collisions, insert the colliding key at the beginning of the linked list. You just need to insert the key and do not have to output anything in this case.
- `d key`: Delete `key` from the hash table. For example, "`d 2`" means "Delete key 2 from the hash table". Do nothing if the hash table does not contain the key. If there are multiple elements with the same key value, delete just one element. If the delete is successful, you have to output:

  ```
  key : DELETED
  ```

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

  ```
  key : DELETE FAILED
  ```

- `s key`: Search `key` in the hash table. If there is an element with the key value, then you have to output:

  ```
  key : FOUND AT i,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 do not find the key, then output:

  ```
  key : NOT FOUND
  ```

- `o`: Output the hash 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.