

Lab04-Hashing

VE281 - Data Structures and Algorithms, Xiaofeng Gao, TA: Qingmin Liu, Autumn 2019

* Please upload your assignment to website. Contact webmaster for any questions.

* Name: _____ Student ID: _____ Email: _____

1. Given a sequence of inputs 192, 42, 142, 56, 39, 319, 14, insert them into a hash table of size 10. Suppose that the hash function is $h(x) = x \% 10$. Show the result for the following implementations:
 - (a) Hash table using separate chaining. Assume that the insertion is always at the beginning of each linked list.
 - (b) Hash table using linear probing.
 - (c) Hash table using quadratic probing.
 - (d) Hash table using double hashing, with the second hash function as $h_2(x) = (x + 4) \% 7$.
2. Show the result of rehashing the four hash tables in the Problem 1. Rehash using a new table size of 14, and a new hash function $h(x) = x \% 14$. (Hint: The order in rehashing depends on the order stored in the old hash table, not on their initial inserting order.)
3. Suppose we want to design a hash table containing at most 900 elements using linear probing. We require that an unsuccessful search needs no more than 8.5 compares and a successful search needs no more than 3 compares on average. Please determine a proper hash table size.
4. Implement queues with two stacks. We know that stacks are first in last out (FILO) and queues are first in first out (FIFO). We can implement queues with two stacks. The method is as follows:
 - For **enqueue** operation, push the element into stack S_1 .
 - For **dequeue** operation, there are two cases:
 - $S_2 = \emptyset$, pop all elements in S_1 , push these elements into S_2 , pop S_2
 - $S_2 \neq \emptyset$, pop S_2

Using amortized analysis to calculate the complexity of **enqueue** and **dequeue** step.