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:Wu Jiayao Student ID:517370910257 Email: jiayaowu1999@sjtu.edu.cn

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 $h_2(x) = (x + 4)\%7$.

Solution.

(a) Separate chaining

[0]	
[1]	
[2]	142 42 192
[3]	
[4]	14
[5]	
[6]	56
[7]	
[8]	
[9]	319 39

Table 1: separate chaining

(b) linear probing

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
319		192	42	142	14	56			39

Table 2: linear probing

(c) quadratic probing

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
319		192	42	14		142	56		39

Table 3: quadratic probing

(d) double hashing

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
56	319	192		14		42		142	39

Table 4: double hashing

□

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.)

Solution.

(a) Separate chaining

[0]	56	14	42
[1]			
[2]	142		
[3]			
[4]			
[5]			
[6]			
[7]			
[8]			
[9]			
[10]	192		
[11]	39	319	
[12]			
[13]			

Table 5: seperate chaining

(b) linear probing

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]
42	14	142	56							192	319	39	

Table 6: linear probing

(c) quadratic probing

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]
42	14	142		56						192	319	39	

Table 7: quadratic probing

(d) double hashing

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]
56		142		14				42		192	319	39	

Table 8: double hashing

□

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.

Solution.

$$U(L) = \frac{1}{2}\left[1 + \left(\frac{1}{1-L}\right)^2\right] \leq 8.5$$

$$S(L) = \frac{1}{2}\left[1 + \frac{1}{1-L}\right] \leq 3$$

Therefore,

$$0 < L \leq \frac{3}{4}$$

$$size \geq 900 \times \frac{4}{3} = 1200$$

The hash table size should be 1201, the smallest prime that meets the requirement. □

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.

Solution.
$\Phi(S)$ denotes the number of items in stack.
One push operation takes:

$$\Phi(S_i) - \Phi(S_i - 1) = 1$$

$$\hat{C}_i = C_i + \Phi(S_i) - \Phi(S_{i-1}) = 2$$

One pop operation takes:

$$\Phi(S_i) - \Phi(S_i - 1) = -1$$

$$\hat{C}_i = C_i + \Phi(S_i) - \Phi(S_{i-1}) = 0$$

(a) **enqueue**
The amortized cost of pushing onto stack 1 is:

$$\hat{C}_i = C_i + \Phi(S_i) - \Phi(S_{i-1}) = 2$$

The amortized cost of poping from stack 1 and pushing onto stack 2 is:

$$\hat{C}_i = (C_i + \Phi(S_i) - \Phi(S_{i-1}))_{push} + (C_i + \Phi(S_i) - \Phi(S_{i-1}))_{pop} = 2$$

One enqueue operation involves the two operations above. The total amortized cost is 4.

(b) **dequeue**

- $S_2 = \emptyset$, the number of items in the stack changes by minus 1.

$$\hat{C}_i = (C_i + \Phi(S_i) - \Phi(S_{i-1}))_{pop} = 1 - 1 = 0;$$

- $S_2 \neq \emptyset$, one pop operation is done

$$\hat{C}_i = (C_i + \Phi(S_i) - \Phi(S_{i-1}))_{pop} = 0;$$

Hence, the amortized cost is 0.

□