# W06 - Chaining - Hashing - Linear Probing - Quadratic Probing results for Ali Asad

(!) Correct answers are hidden.

Score for this attempt: 20 out of 20

Submitted 24 Feb at 0:29

This attempt took 53 minutes.

The following questions refer to a hash table with the following properties.

- the maximum load factor is 0.75
- the minimum load factor is 0.33
- the backing array has a size of at least 2
- the hash function is  $h(x) = (x-3) \mod N$ , where N is the size of the backing array.
- the collision resolution strategy is chaining, i.e. colliding elements are chained into a list at the colliding index of the backing array.
- the hash table is initially empty and its backing array has size 2.

Positive numbers are added and removed to this hash table as follows:

A + next to a number indicates that the number is *add*ed to the table, and a - indicates that it is *remove*d from the table. The operations are applied in order from left to right.

Question 1 1/1 pts

Once the operations are complete, how many elements does the hash table contain?

4

### **Question 2**

1 / 1 pts

Once the operations are complete, what is the length of the longest chain?

2

### **Question 3**

1 / 1 pts

Once the operations are complete, what is the size of the backing array?

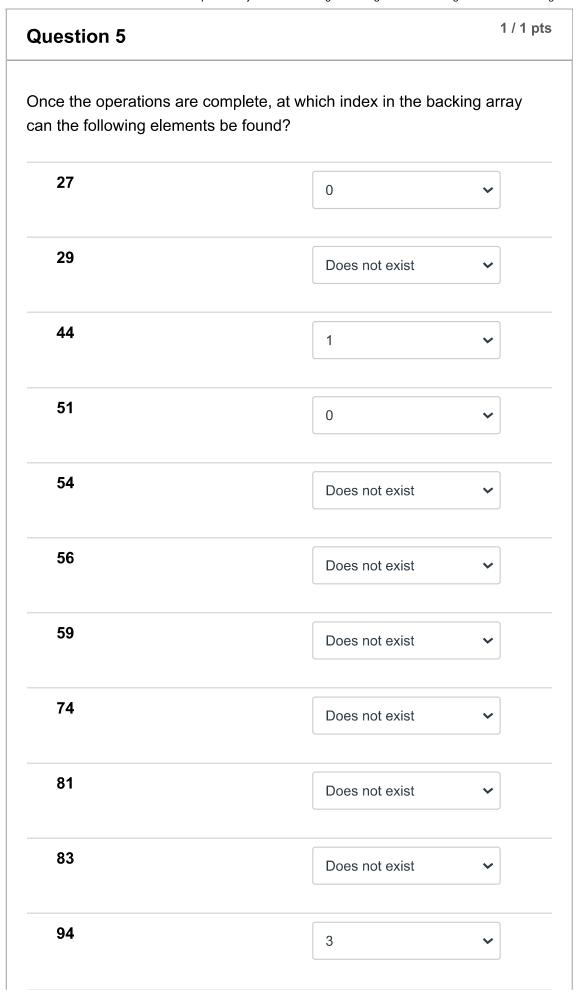
8

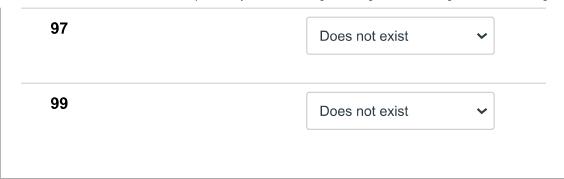
### **Question 4**

1 / 1 pts

Once the operations are complete, how many indexes in the backing array are unoccupied?

5





# 1 / 1 pts **Question 6** For efficient use of memory, a hash table may maintain a maximum and a minimum load factor. The hash table is resized to a size equal to twice the number of its elements if the occupancy exceeds the maximum load factor or falls below the minimum load factor. A maximum load factor that is less than 1 implies that additional memory will be reserved even though some of the currently reserved memory is still unused. Why would a hash table maintain a maximum load factor less than 1? When the maximum load factor is exceeded, potential collisions may be degrading the performance of the hash table. When the maximum load factor is exceeded, there are more elements to store than the amount of reserved memory. When the maximum load factor is exceeded, there is no more space to store further elements. When the maximum load factor is exceeded, the currently reserved memory is all occupied.

Question 7	1 / 1 pts
Consider two functions,	
$f(x)=x\mod N,\ g(x)=x^2\mod N$	
What could be some reasons to prefer $g(x)$ over $f(x)$ as a hash function?	l
Mark all that apply.	
f(x) generally maps consecutive elements to consecutive indexes.	
g(x) is more mathematically complex.	
g(x) takes more time to compute.	
g(x) hashes more randomly compared to $f(x)$ .	

Question 8 1/1 pts

The website, <a href="https://www.random.org/">https://www.random.org/</a>, <a href="https://www.random.org/">https://www.random.org/</a>, <a href="https://www.random.org/">generates random numbers based on atmospheric noise. Imagine a hash function that hashes x as follows,</a>

 $h(x) = x * ext{random number from random.org} \mod N$  Which of the following statements about the hash function are true?

Mark all that apply.

<b>✓</b>	It is a good	hash function	on because	the output	is random.
----------	--------------	---------------	------------	------------	------------

It is a bad has function because it uses the modulo operation.

- It is a good hash function as it is efficient to compute,
- It is a bad hash function because the hash value is not deterministic.

The following questions refer to a hash table with the following properties.

- the maximum load factor is 0.75
- the minimum load factor is 0.33
- the backing array has a size of at least 2
- the hash function is  $h(x)=(x-3)\mod N$ , where N is the size of the backing array.
- the collision resolution strategy is linear probing, i.e. the  $i^{ ext{th}}$  probe for x is at the index,  $(h(x)+i) \mod N$ , in the backing array which has size N.
- DEL elements are counted in the occupancy when comparing with the maximum load factor.
- DEL elements are not counted in the occupancy when comparing with the minimum load factor.
- the hash table is initially empty and its backing array has size 2.

Positive numbers are added and removed to this hash table as follows:

A + next to a number indicates that the number is *add*ed to the table, and a - indicates that it is *remove*d from the table. The operations are applied in order from left to right.

Question 9 1/1 pts

Once the operations are complete, how many elements does the hash table contain?

6

# **Question 10**

1 / 1 pts

Once the operations are complete, how many DEL elements does the hash table contain?

0

### **Question 11**

1 / 1 pts

Once the operations are complete, what is the size of the backing array?

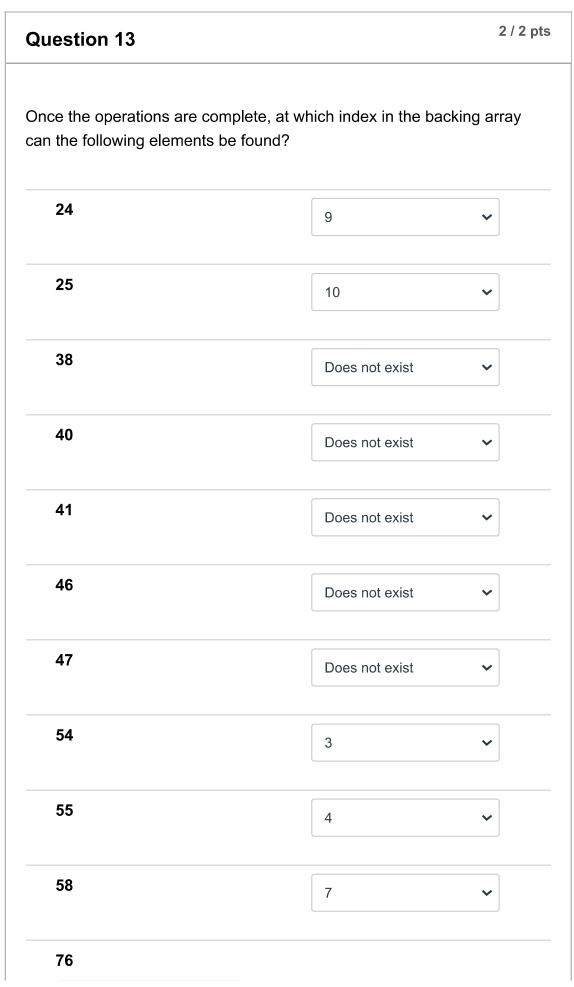
12

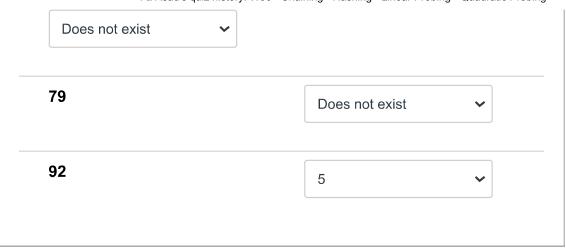
## **Question 12**

1 / 1 pts

Once the operations are complete, how many indexes in the backing array are unoccupied?

6





The following questions refer to a hash table with the following properties.

- the maximum load factor is 0.75
- the minimum load factor is 0.33
- the backing array has a size of at least 2
- the hash function is  $h(x) = (x-3) \mod N$ , where N is the size of the backing array.
- the collision resolution strategy is linear probing, i.e. the  $i^{\text{th}}$  probe for x is at the index,  $(h(x) + i^2) \mod N$ , in the backing array which has size, N.
- DEL elements are counted in the occupancy when comparing with the maximum load factor.
- DEL elements are not counted in the occupancy when comparing with the minimum load factor.
- the hash table is initially empty and its backing array has size 2.

Positive numbers are added and removed to this hash table as follows:

A + next to a number indicates that the number is *add*ed to the table, and a - indicates that it is *remove*d from the table. The operations are applied in order from left to right.

**Question 14** 

1 / 1 pts

table contain?	
3	

Once the operations are complete, how many DEL elements does the hash table contain?

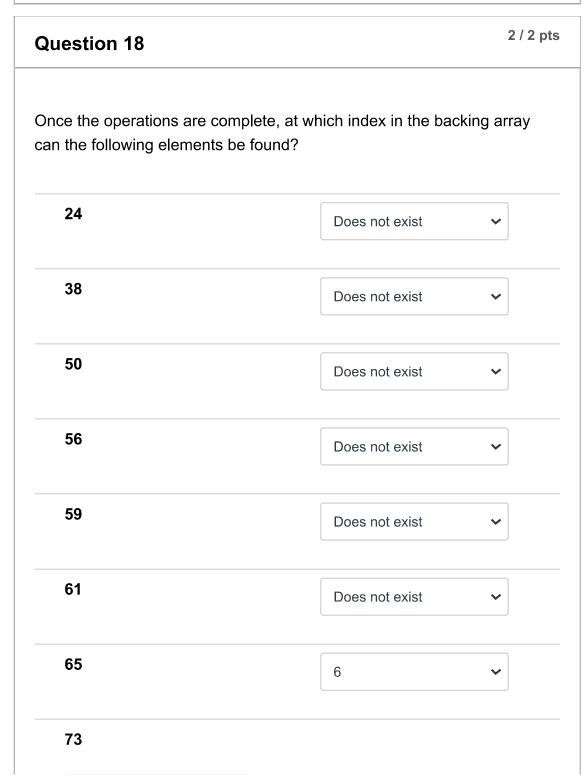
Question 16

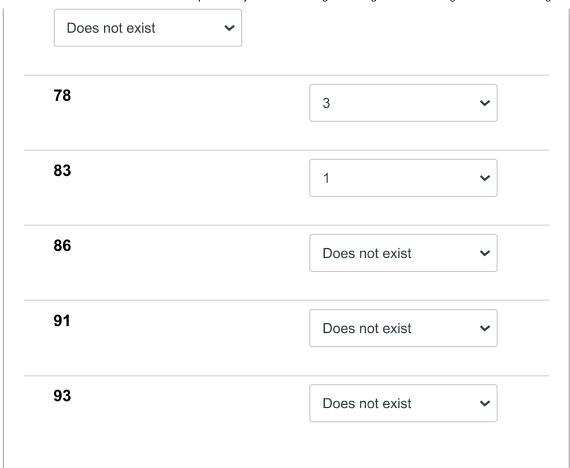
Once the operations are complete, what is the size of the backing array?

Question 17

1 / 1 pts

Once the operations are complete, how many indexes in the backing array are unoccupied?





Quiz score: 20 out of 20