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:

+59, -59, +99, +56, -56, -99, +97, -97, -29, -54, -74, -81, +44, +99, +99, +27, +94, -83, -99, +51

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

Question 5

1 / 1 pts

Once the operations are complete, at which index in the backing array can the following elements be found?

27	0
29	Does not exist
44	1
51	0
54	Does not exist
56	Does not exist
59	Does not exist
74	Does not exist
81	Does not exist
83	Does not exist
94	3

97	Does not exist
99	Does not exist

Question 6

1 / 1 pts

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?

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, **https://www.random.org/,** **(https://www.random.org/,)** generates random numbers based on atmospheric noise. Imagine a hash function that hashes x as follows,

$$h(x) = x * \text{ random number from random.org } \mod N$$

Which of the following statements about the hash function are true?

Mark all that apply.

☑ It is a good hash function 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^{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:

+25, -47, +46, +38, +79, +41, -38, +40, -79, +55, +58, -41, +47, +24, -47, +54, -76, -46, -40, +92

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

Question 13

2 / 2 pts

Once the operations are complete, at which index in the backing array can the following elements be found?

24	9
25	10
38	Does not exist
40	Does not exist
41	Does not exist
46	Does not exist
47	Does not exist
54	3
55	4
58	7
76	

Does not exist

79	Does not exist
92	5

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^{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:

-86, -56, +50, +83, -83, -50, -91, -61, +73, -73, +59, +78, -24, +83, +93, +38, -59, -93, +65, -38

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

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

3

Question 15

1 / 1 pts

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

3

Question 16

1 / 1 pts

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

8

Question 17

1 / 1 pts

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

2

Question 18

2 / 2 pts

Once the operations are complete, at which index in the backing array can the following elements be found?

24	Does not exist
38	Does not exist
50	Does not exist
56	Does not exist
59	Does not exist
61	Does not exist
65	6
73	

Does not exist

78	3
83	1
86	Does not exist
91	Does not exist
93	Does not exist

Quiz score: **20** out of 20