**HashChainApp.java:**

**Investigate how the load factor affects the average probe length.**

Sample Output:  
Enter size of hash table: 10

Enter initial number of items: 5

Inserted key: 482 with probe length: 1

Inserted key: 404 with probe length: 1

Inserted key: 93 with probe length: 1

Inserted key: 151 with probe length: 1

Inserted key: 424 with probe length: 2

Initial keys: 482 404 93 151 424

Average probe length for initial filling: 1.2

Explanation:

In the output, I have inputted a hash table with the size of 10 and inserted 5 initial items. The load factor is calculated as the ratio of the number of items inserted to the size of the hash table. In this case, the load factor is = 0.5.

The average probe length for the initial filling of the table is 1.2. This means that, on average, it took 1.2 probes to insert each item into the hash table.

In general, as the load factor increases, the average probe length tends to increase as well. This is because a higher load factor means more items are being stored in fewer buckets, leading to more collisions and longer probe lengths. However, the relationship between load factor and probe length can be influenced by various factors, including the hash function, the distribution of keys, and the collision resolution strategy.

**HashTableApp.java**

**Investigate how the load factor affects the average probe length.**

Sample Output:

Enter size of hash table: 10

Hash value: 3 | Probe sequence: 3 | Probe length: 1

Load factor: 0.1 | Average probe length: 1.0

Hash value: 4 | Probe sequence: 4 | Probe length: 1

Hash value: 0 | Probe sequence: 0 | Probe length: 1

Load factor: 0.25 | Average probe length: 1.0

Hash value: 1 | Probe sequence: 1 | Probe length: 1

Hash value: 2 | Probe sequence: 2 | Probe length: 1

Hash value: 2 | Probe sequence: 2 -> 3 | Probe length: 2

Hash value: 8 | Probe sequence: 8 | Probe length: 1

Hash value: 5 | Probe sequence: 5 | Probe length: 1

Load factor: 0.5 | Average probe length: 1.2

Hash value: 8 | Probe sequence: 8 | Probe length: 1

Hash value: 3 | Probe sequence: 3 | Probe length: 1

Hash value: 3 | Probe sequence: 3 -> 4 | Probe length: 2

Hash value: 3 | Probe sequence: 3 -> 4 -> 8 -> 7 | Probe length: 4

Hash value: 3 | Probe sequence: 3 -> 4 -> 8 -> 7 -> 3 -> 8 -> 4 -> 3 -> 7 -> 8 -> 8 -> 9 | Probe length: 12

Hash value: 9 | Probe sequence: 9 -> 0 | Probe length: 2

Hash value: 8 | Probe sequence: 8 -> 9 -> 3 -> 2 | Probe length: 4

Load factor: 0.75 | Average probe length: 3.7142857142857144

Hash value: 4 | Probe sequence: 4 | Probe length: 1

Hash value: 0 | Probe sequence: 0 | Probe length: 1

Hash value: 4 | Probe sequence: 4 -> 5 | Probe length: 2

Hash value: 2 | Probe sequence: 2 | Probe length: 1

Hash value: 6 | Probe sequence: 6 | Probe length: 1

Hash value: 6 | Probe sequence: 6 -> 7 | Probe length: 2

Hash value: 9 | Probe sequence: 9 | Probe length: 1

Hash value: 3 | Probe sequence: 3 | Probe length: 1

Hash value: 0 | Probe sequence: 0 -> 1 | Probe length: 2

Load factor: 0.9 | Average probe length: 1.3333333333333333

Enter first letter of show, insert, delete, find, or quit: 5

Invalid entry

Enter first letter of show, insert, delete, find, or quit: s

Table: \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\*

Enter first letter of show, insert, delete, find, or quit: q

Quitted.

Explanation:

In the provided output, the load factor is varied from 0.1 to 0.9, and for each load factor, the average probe length is calculated and displayed.

* **Load factor 0.1**: With a low load factor, there are fewer items stored in the hash table compared to its size. Hence, there are fewer collisions, and the average probe length remains low (1.0 in this case).
* **Load factor 0.25**: As the load factor increases, more items are stored in the hash table. However, the table size still provides sufficient space, resulting in a low average probe length (1.0).
* **Load factor 0.5**: At this load factor, the number of items stored is half the size of the table. Some collisions start occurring, leading to a slightly higher average probe length (1.2).
* **Load factor 0.75**: With a higher load factor, more items are stored, and collisions become more frequent. This results in a considerable increase in the average probe length (3.714).
* **Load factor 0.9**: At this load factor, the table is almost fully occupied, leading to many collisions. Consequently, the average probe length increases further, but it's slightly lower than the load factor of 0.75 (1.333).

In summary, as the load factor increases, the average probe length tends to increase as well. This is because higher load factors lead to more collisions, requiring more probes to find suitable positions for insertion. However, other factors such as the hash function and collision resolution strategy also influence the relationship between load factor and average probe length.