

# Answers to End of Section and Review Exercises for Chapter 11

## Exercises 11.1

1. The items in a list are ordered by position, whereas in a set they are in no particular order. The items in a set are unique, whereas a list can contain duplicate items.
2. Answers:
  - a. {3, 4}
  - b. {3, 4}
  - c. {3, 4, 5}
  - d. {4, 5}
3. You visit all items in a set by using a `for` loop over the set.

## Exercises 11.2

1. During a union, all the items from both sets are added to a third set. The traversal of each set is linear, and the addition of each item to the third set is also linear. Thus, the running time of union is approximately  $O(n^2)$ . The performance of difference and intersection is similar.
2. If the set is not searched during an addition, the addition can run in constant time on the average. However, extra memory is required to hold duplicate items, and all instances of an item must be removed when that item is removed. Finally, a traversal needs to skip duplicate items.

## Exercises 11.3

1. Hashing uses a constant-time function to transform a key or item into an index position in an array. If that position is unoccupied, the hashing strategy works perfectly.
2. A home index is the position to which an item first hashes.
3. Collisions occur when two keys or items hash to the same index position.
4. The linear method resolves collisions by searching for the first empty index position in the array beyond the index to which an item or key is hashed.
5. Clustering results from several collisions. Data are inserted at adjacent positions in the array. This leads to further collisions and other clusters, which eventually coalesce into larger clusters.
6. Unlike the linear method of resolving collisions, which advances through the array by incrementing an index by 1, the quadratic method advances through the array by incrementing an index by the square of the previous increment value. The quadratic method thus has the potential to move the index over a cluster of

- occupied positions with a single increment, thereby increasing the likelihood of fewer iterations in the search for an empty position.
7. Load factors:
    - a. .3
    - b. 1
    - c. 3.3
  8. Chaining is a method to resolve collisions. The occupied index positions contain linked structures of items. When a collision occurs, the item is added to the head of the linked structure at that position.

## Exercises 11.4

The implementing class computes the load factor after each insertion of an item. If this factor is greater than 0.75, the method creates a new array twice the capacity of the old array, rehashes the items into the new array, and resets the `self._items` instance variable to the new array.

## Answers to Review Questions

1. c
2. b
3. b
4. a
5. b
6. c
7. b
8. c
9. b
10. b