**SET**

The classes that implement the List interface are all indexed collections

* An index or subscript is associated with each element
  + We can talk about next object, previous object
* The element’s index often reflects the relative order of its insertion into the list
* Searching for a particular value in a list is generally O(n)
* An exception is a binary search of a sorted object, which is O(logn)

In this chapter, we consider another part of the Collection hierarchy: the Set interface and the classes that implement it

Set objects:

* are not indexed
* do not reveal the order of insertion of items
* enable efficient search and retrieval of information
* allow removal of elements without moving other elements around

Diagram

Description automatically generated

**MAP**

Relative to a Set, Map objects provide efficient search and retrieval of entries that contain pairs of objects (a unique key and the information)

Hash tables (implemented by a Map or Set) store objects at arbitrary locations and offer an average constant time for insertion, removal, and searching