D:\Users\A0344\Desktop\collection-hierarchy.png

Collection Map

/ / \ \ |

/ / \ \ |

Set List Queue Dequeue SortedMap

/

/

SortedSet

**Core Interfaces in Collections**

Note that this diagram shows only core interfaces.

**Collection :** Root interface with basic methods like add(), remove(),

contains(), isEmpty(), addAll(), ... etc.

**Set :** Doesn't allow duplicates. Example implementations of Set

interface are HashSet (Hashing based) and TreeSet (balanced

BST based). Note that TreeSet implements **SortedSet**.

**List :** Can contain duplicates and elements are ordered. Example

implementations are LinkedList (linked list based) and

[ArrayList](https://www.geeksforgeeks.org/array-vs-arraylist-in-java/) (dynamic array based)

**Queue :** Typically order elements in FIFO order except exceptions

like PriorityQueue.

**Deque :** Elements can be inserted and removed at both ends. Allows

both LIFO and FIFO.

**Map :** Contains Key value pairs. Doesn't allow duplicates. Example

implementation are [HashMap and TreeMap](https://www.geeksforgeeks.org/hashmap-treemap-java/).

TreeMap implements **SortedMap**.

The difference between Set and Map interface is, in Set we have only

keys, but in Map, we have key value pairs.

