Page No.:

Sorted Set(I):-

- It is the child interface of Set.
- If we want to represent a group of individual objects as a single entity where duplicates are not allowed and all objects should be inserted according to some sorting order then we should go for Sorted Set.

→ Navigable Set (I):-

- It is the child interface of Sorted Set.
- It contains several methods for Navigation purposes.

⇒ Diff b/w List and Set:→

List	Set
(i) Duplicates ✓	(i) ✗
(ii) Insertion order preserved.	(ii) not preserved.

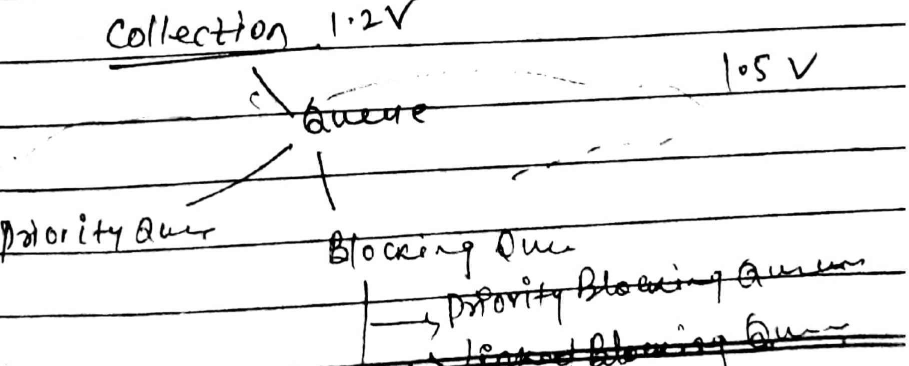
→ Queue Interface(I):→

→ It is the child interface of collection.

→ If we want to represent a group of individual objects prior to processing, then we should go for Queue.

→ Usually Queue follows first in first order (FIFO) but based on our requirement, we can implement our own priority order also.

Ex:- Before sending a mail, All mail id's, we have to store in some data structure, in which order we added mail id, in the same order only mail should be delivered. For this requirement, Queue is based choice.



Part-2

→ NOTE :—

All the above interfaces (collection, List, Sorted set, Set, NavigableSet and Queue) meant for representing a group of individual Objects.
If we want to represent, a group of objects as key-value pairs then we should go for Map (I).

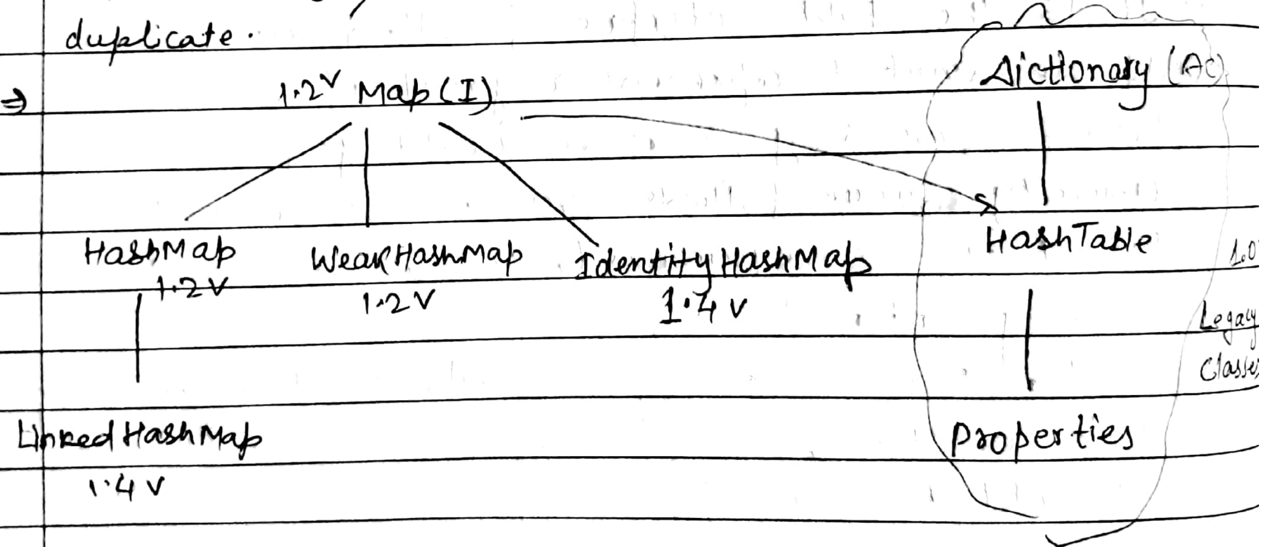
→ Map :—

→ Map is not child interface of collection.
→ If we want to represent a group of objects as key-value pairs then we should go for Map.

→

Key	Value
Roll No	Student Name
101	durga
102	Ravi
103	Shiva

→ Both key and value are objects only.
→ duplicate keys are not allowed but values can be duplicate.

→


→ SortedMap :— It is the child interface of Map (I).
→ If we want to represent, a group of key-value pairs according to some sorting order of keys then we should go for Sorted Map.
→ In sorted Map, the sorting should be based on key but not based on value.