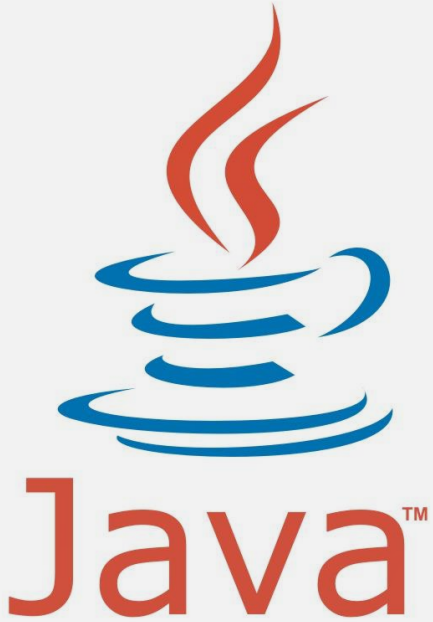
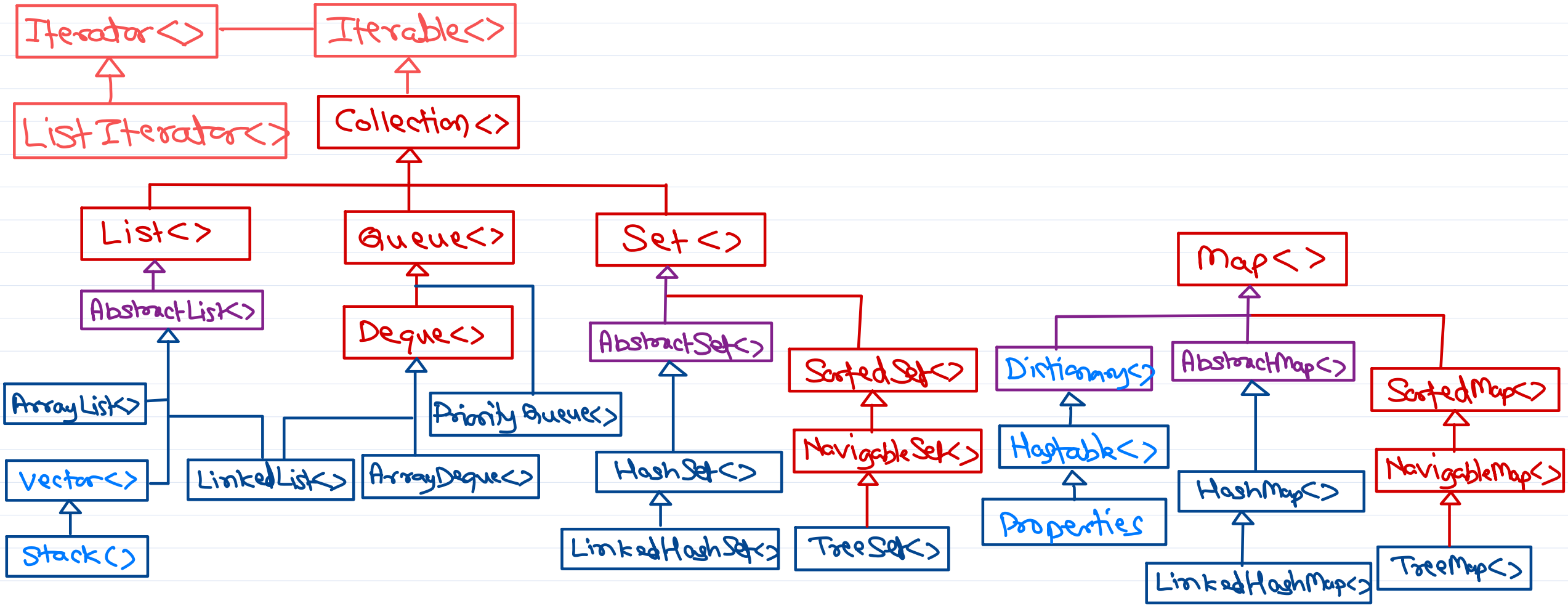


Core Java

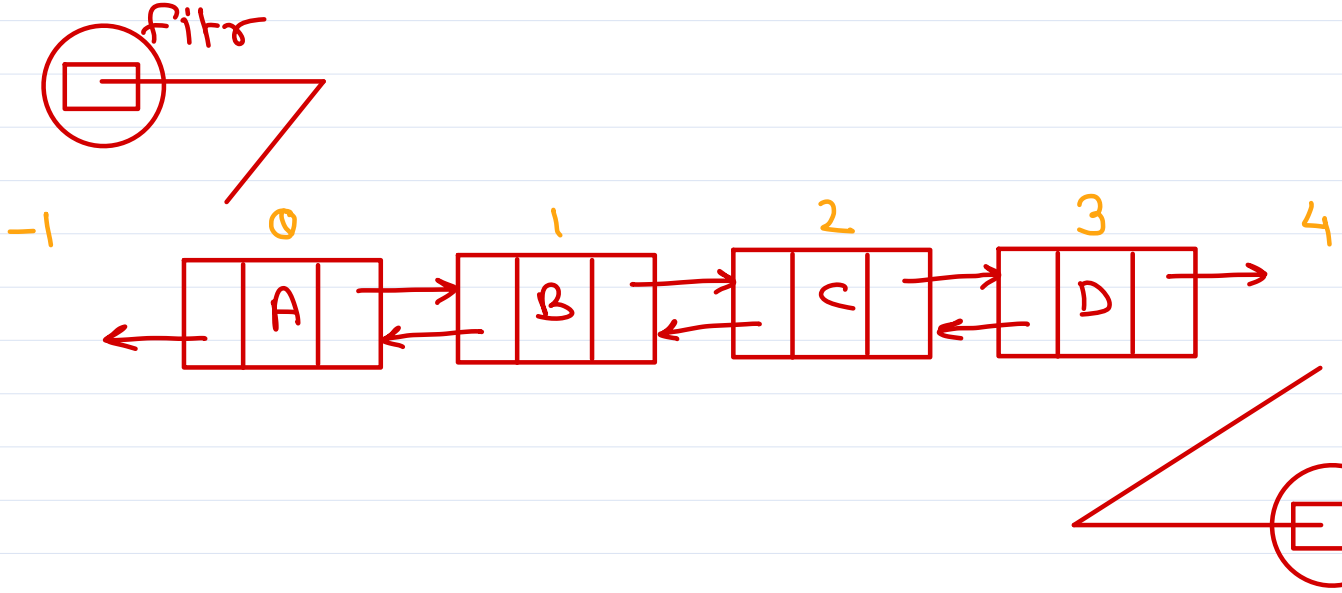
Trainer: Nilesh Ghule

Java Collection Framework



Forward and Reverse traversal using ListIterator

hasNext() → check if ele present at cur pos.
next() → return ele at cur pos & go to next pos.




```
fitr = list.listIterator();
while(fitr.hasNext()) {
    ele = fitr.next();
    ...
```

3

```
ritr = list.listIterator(list.size());
while(ritr.hasPrevious()) {
    ele = ritr.previous();
    ...
```

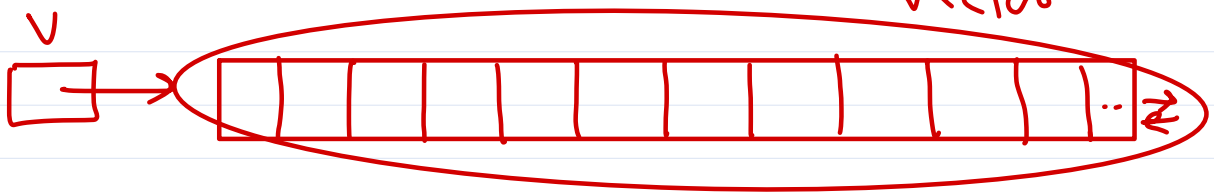
ritr 3

hasPrevious() – check if ele present at prev pos.
previous() – go to prev pos & return ele at that pos.

Vector vs ArrayList vs LinkedList

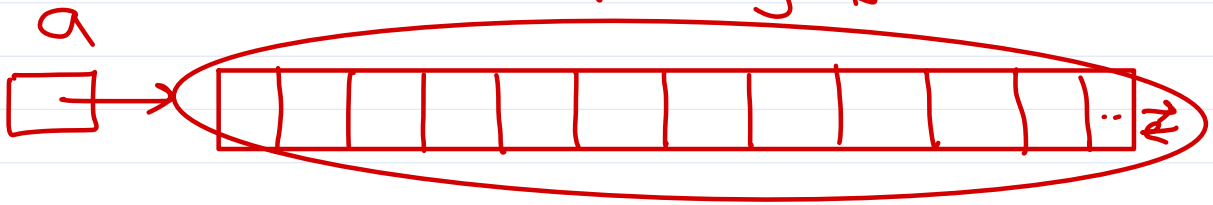
Vector v = new Vector();

* Dynamically growable/shrinkable array.
* Synchronized class / slower
* Legacy (1.0) * growth = 2 * capacity

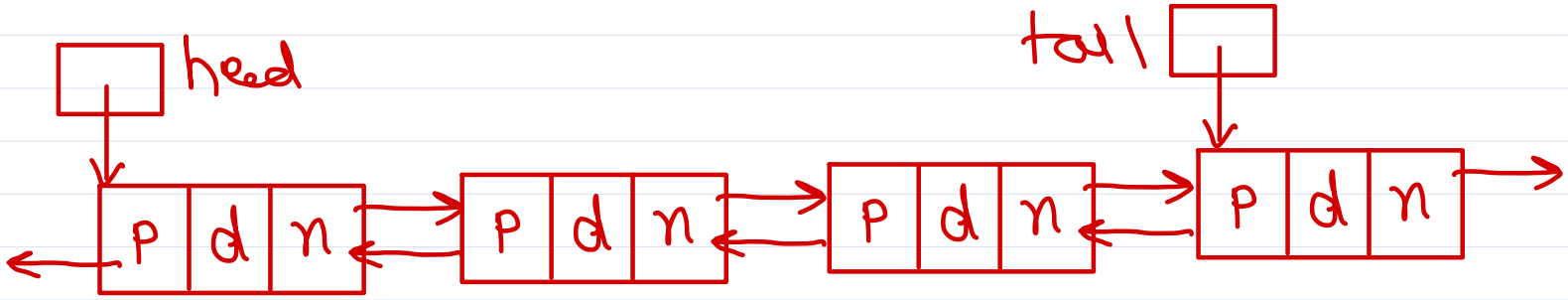


ArrayList a = new ArrayList();

* Dynamically growable/shrinkable array - fast random access.
* Non-synchronized / Faster
* Collection framework (1.2)
* growth = 1.5 * capacity

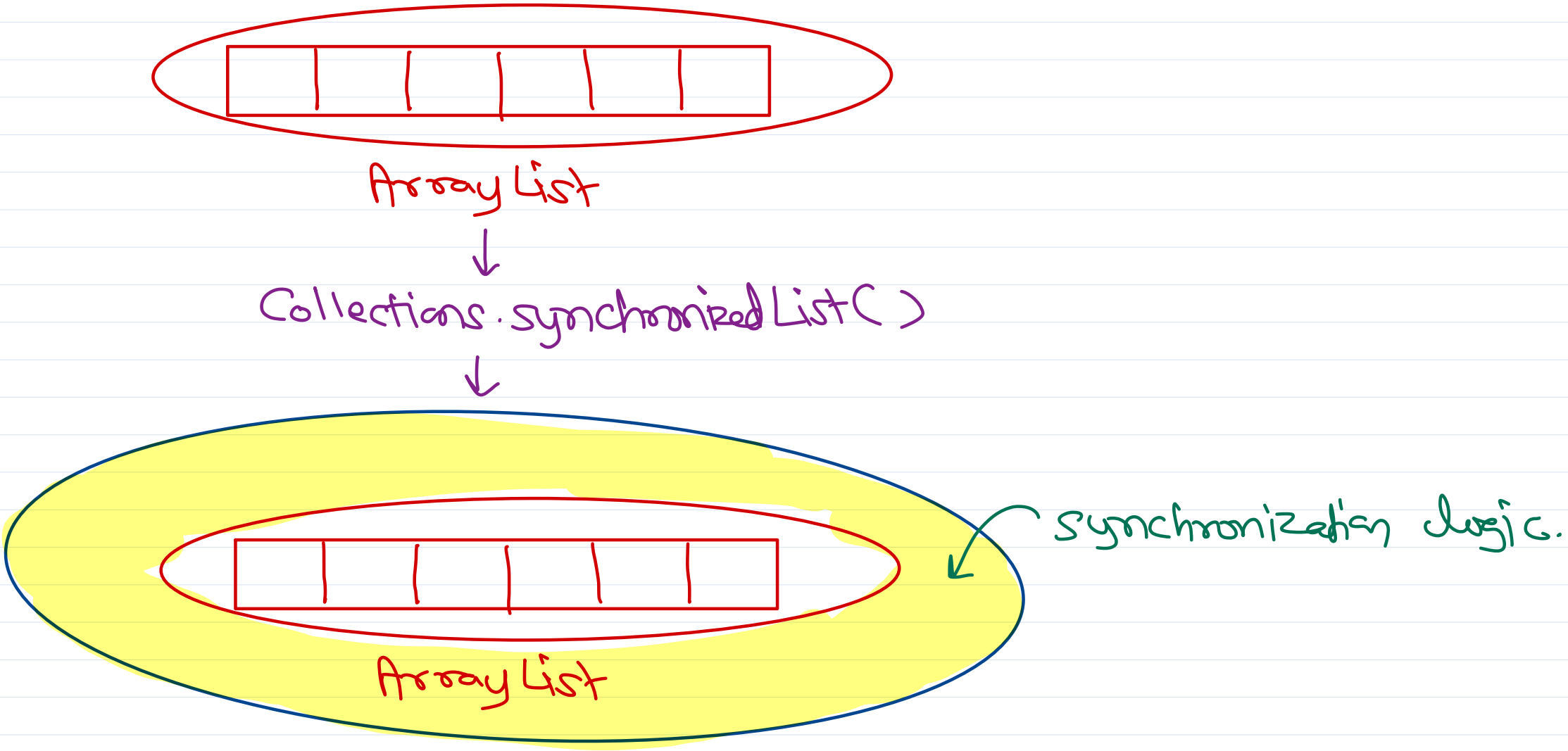


LinkedList l = new LinkedList();

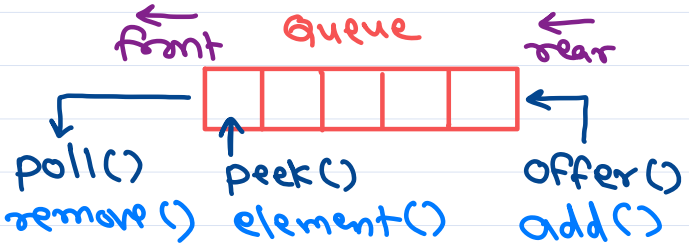


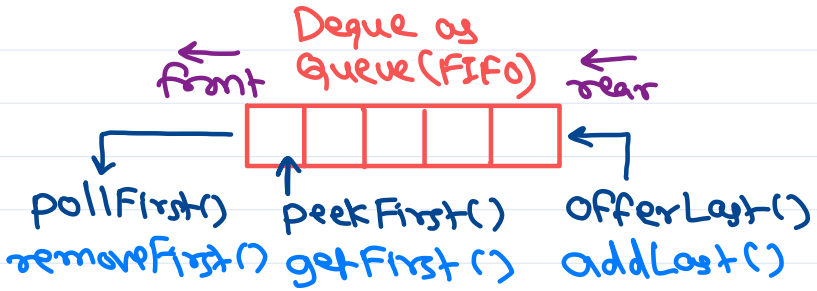
* Doubly Linked List
* frequent add/delete op.
* slower random access.
* Inherited from List, Queue

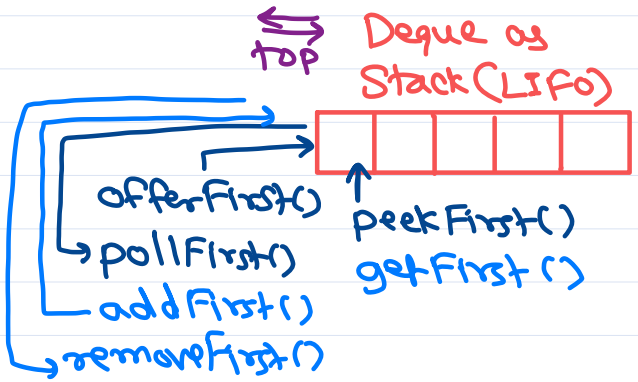
Collections.synchronizedList()



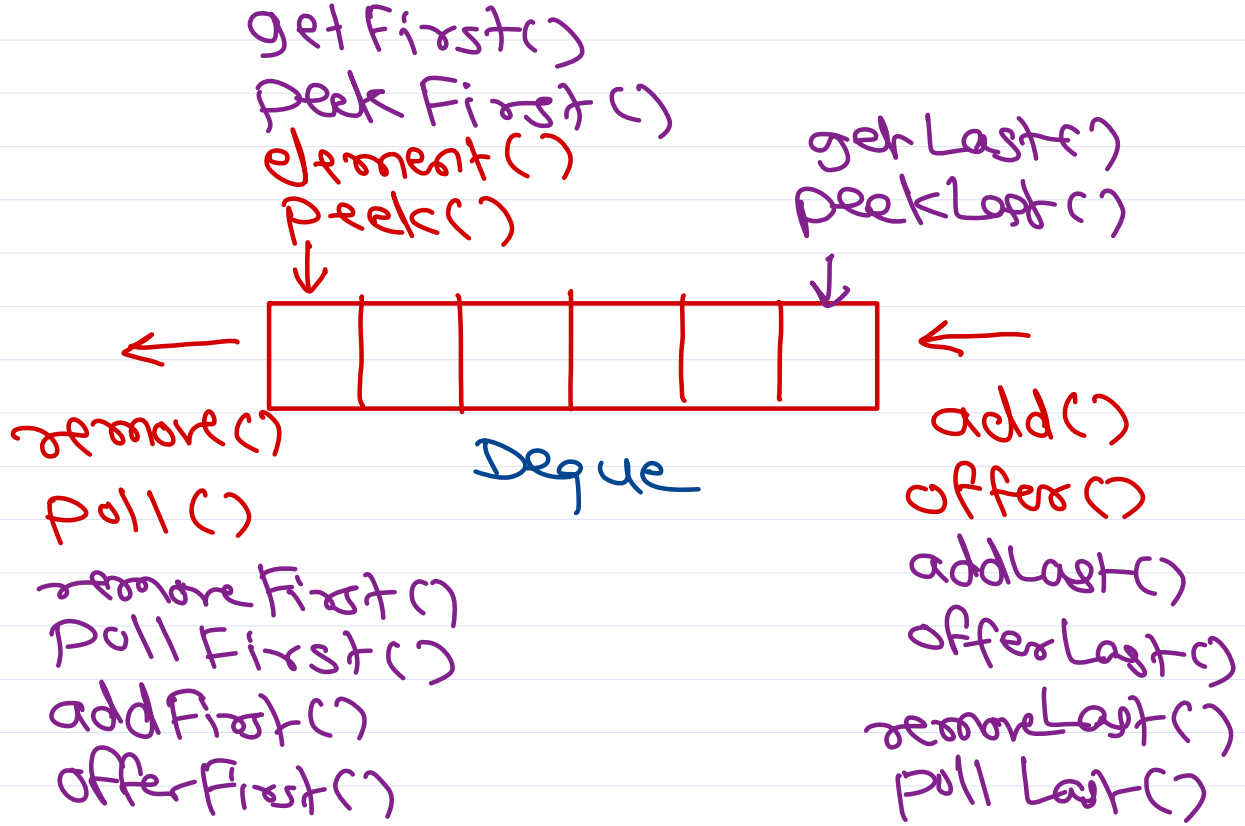
Queue







Queue D.S.	Stack D.S.
① FIFO	① LIFO
② Add/Remove is from Diff ends (Rear/Front)	② Add/Remove is from Same end (top).



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>