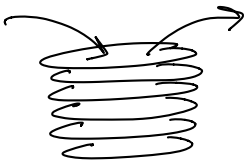
List ADT

- any sort of linear structure
  - Implementation: Dynamically-growing array
    ArrayLists, RArrayList, Python lists
  - Implementation: Linked Lists
    - Singly-linked lists
    - Doubly-linked lists

Stacks + Queues – ADTs

- restricted versions of List ADT

Stack – ADT

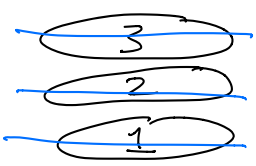


- **Push**: Add an element to the top of the stack.
- **Pop**: Removes & returns the top element of the stack.
- Peek: Returns the top element but doesn't remove it.
- isEmpty(): returns true if the stack is empty.

Stack – push & pop always happen @ the same end of the list.

Stack of integers

→ Push 1
→ Push 2
→ Push 3
Pop
Pop
Pop

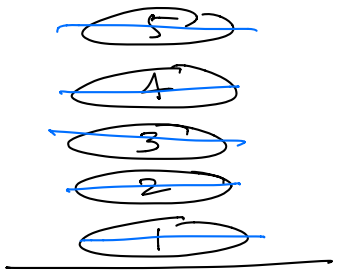


ORDER OF POPPING
3
2
1

"LIFO" – last in, first out

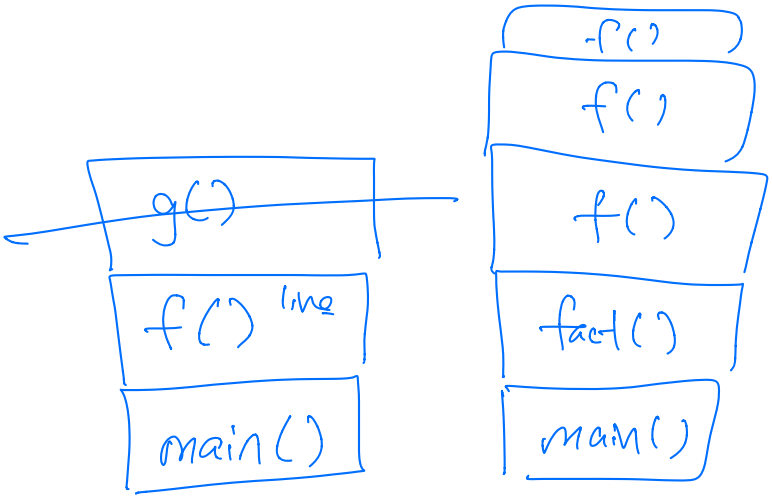
Ex

Push 1
Push 2
Push 3
Pop
Push 4
Pop
Pop
Push 5
Pop
Pop



ORDER

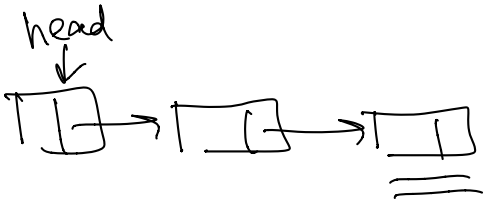
3 ←
4 ←
2 ←
5 ←
1 ←



( )
{ }

((

Stack op	Array list	Linked list (SLL)
Push	insert @ right O(1)	insert @ head O(1)
Pop	delete from right O(1)	delete @ head O(1)

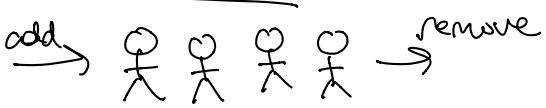


Apps

Web browser history
Undo history

Queues - ADT

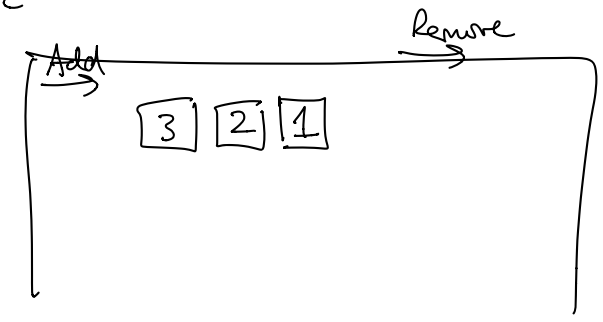
- waiting line



- ENQUEUE / ADD / OFFER
- DEQUEUE / REMOVE

Enqueue 1
" 2
" 3

Dequeue
"
"



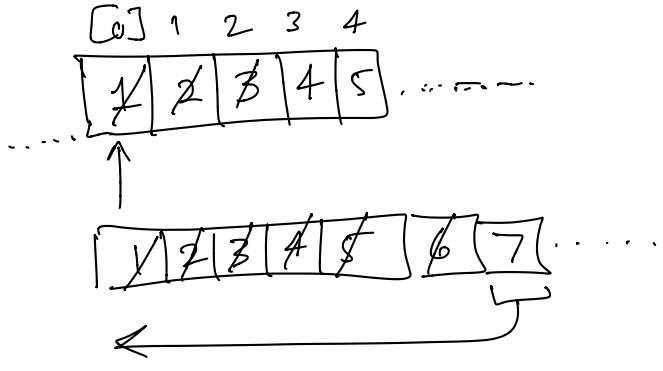
ORDER
1
2
3

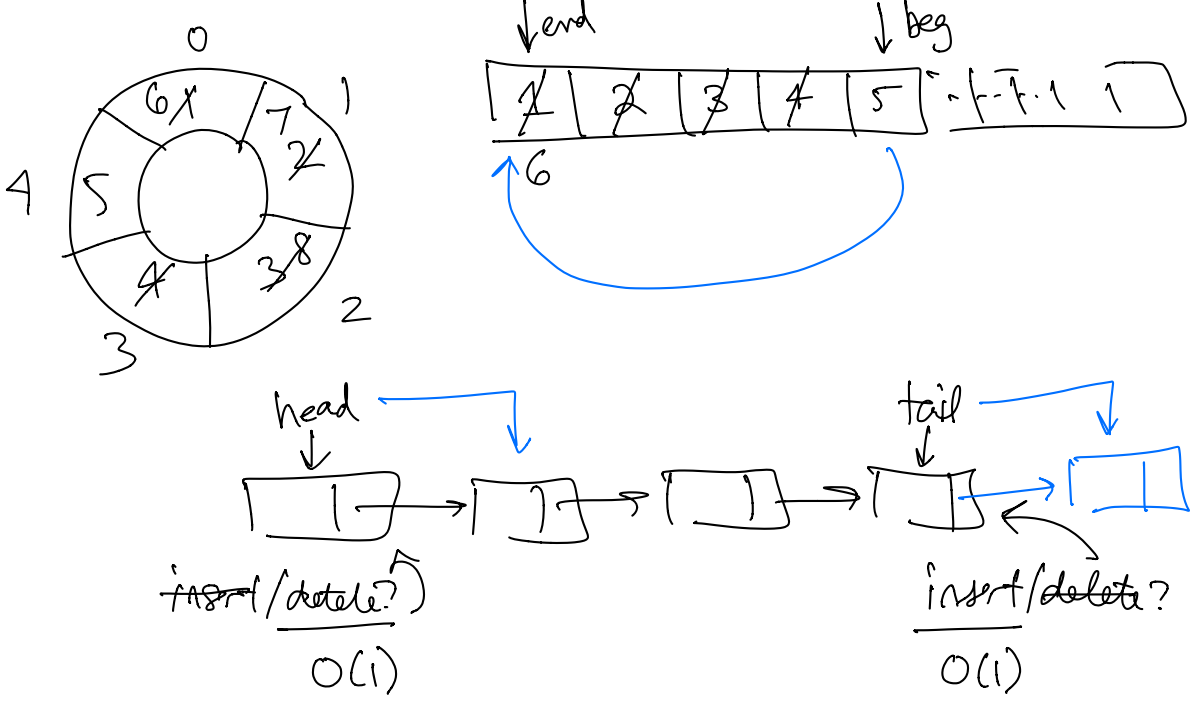
Enq 1 ✓
" 2 ✓
" 3 ✓
Deq
Enq 4
Deq
Deq
Enq 5

ORDER
1
2
3
4
5

Implementations

ArrayList




0
1
2
3
4
5
6
7
8
end
beg
6

head
tail
insert/delete?
O(1)
insert/delete?
O(1)