

Integer Queue : instance

<u>Operation</u>	<u>state</u>	<u>return</u>
Constructor $\longrightarrow$	()	-
Enqueue(5)	(5)	-
Enqueue(1)	(5, 1)	-
Enqueue(7)	(5, 1, 7)	-
Dequeue()	(1, 7)	-
Enqueue(3)	(1, 7, 3)	-
getLength()	(1, 7, 3)	3
⋮	⋮	

note: Some ops are undefined on certain states

- getFront() on empty state ()
- Dequeue() " " " "

Don't make up default behaviors ;
we establish preconditions for each ADT op.

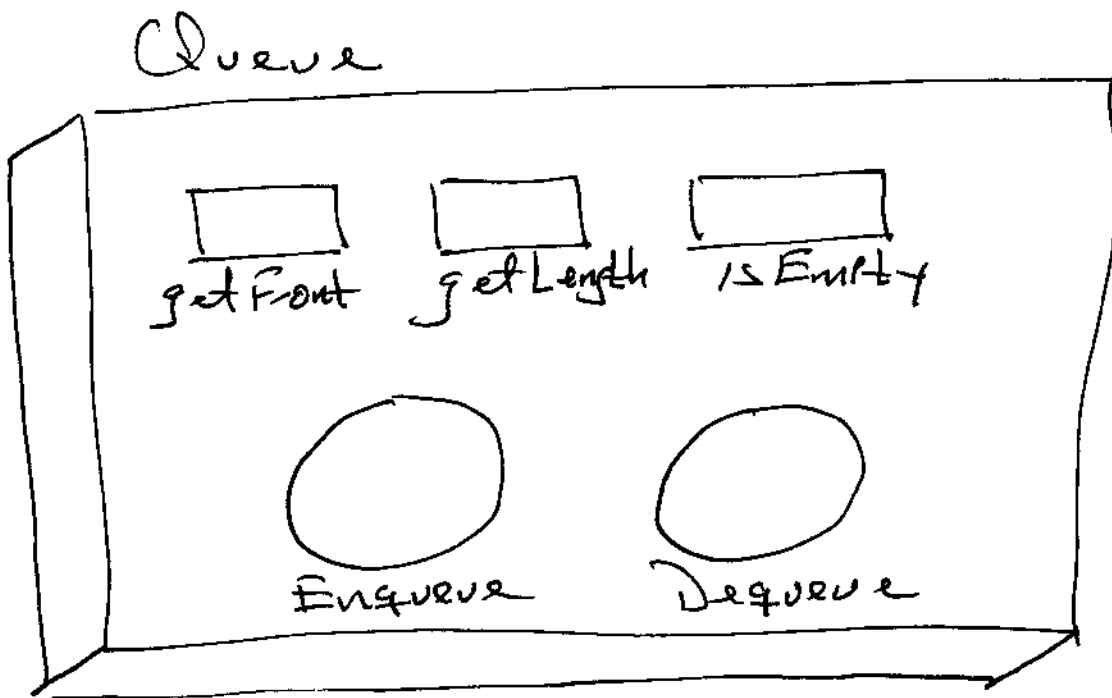
Preconditions are stated in terms of access function calls.

Policy:

If an ADT op. is called on a state that violates its preconditions, it should print an error message giving

ADT: OP: Precond,
then quit program.

Black Box Picture (information hiding)



Pres. not is Empty()

for both getFront() & Dequeue()

To specify an ADT must give

- states
- OPS
- Precond. to all OPS,

Test ADT or pre cond. like this.

operation {

if (Pre.cond. violated) {

error message

exit.

}

if (Pre. cond. violated) {

—

—

}

nominal execution

⋮

}

ADTs vs. Data structures

ADT

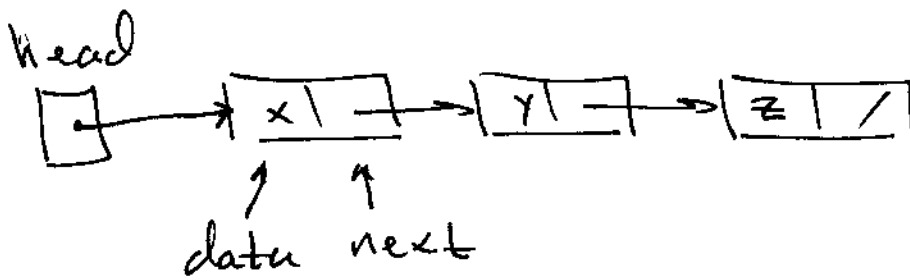
Data Structures.

array

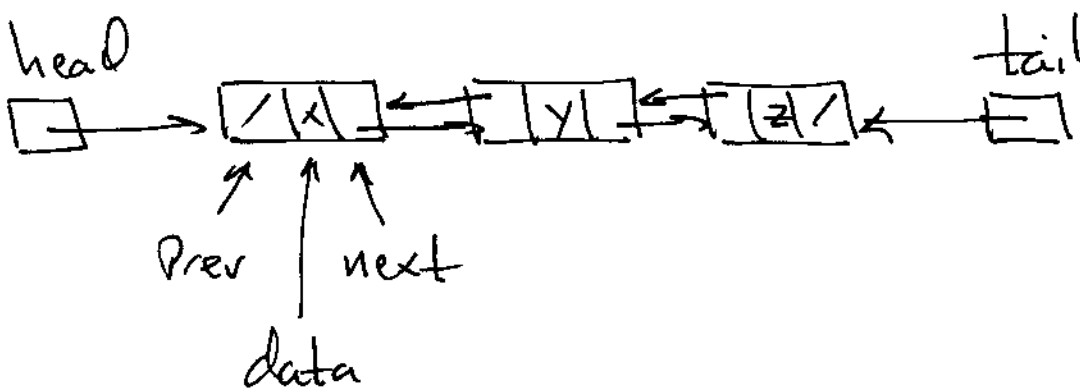
linked list

skip list

singly linked list :



doubly linked list :



skip list : hierarchy of linked lists .

