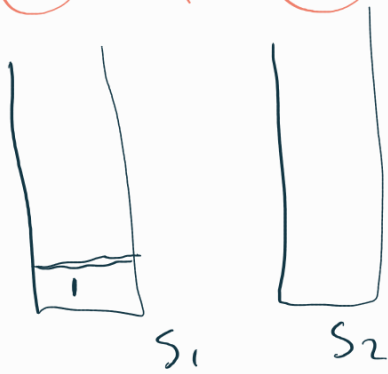
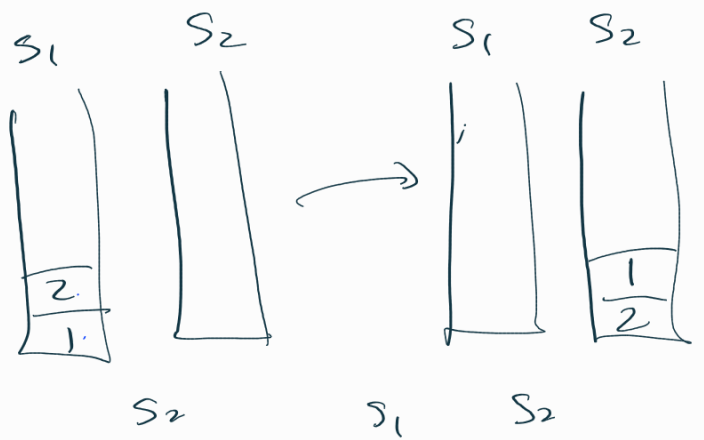


1st $S_1 \rightarrow \text{enqueue}$ $S_2 \rightarrow \text{dequeue}$

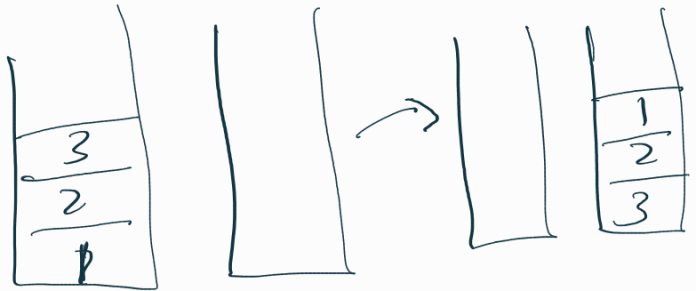


insert(2)

$O(1)$

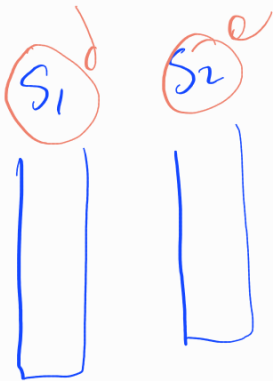


insert(3)

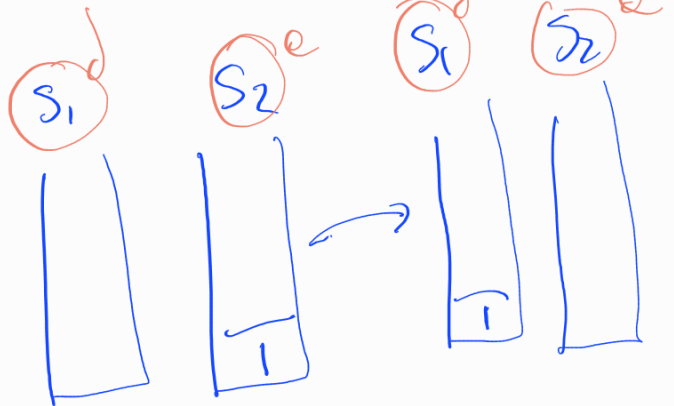


FIFO

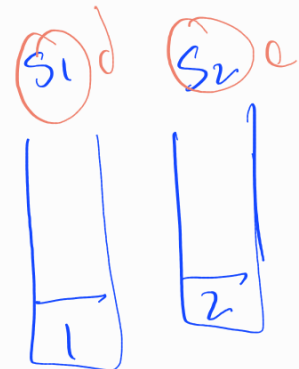
2nd



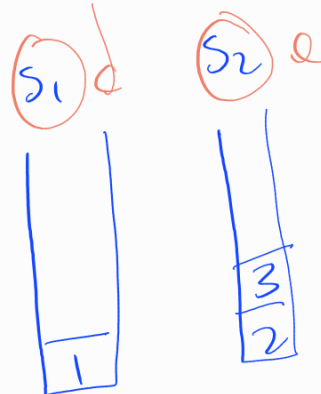
insert(i)
 $O(1)$



insert(2)



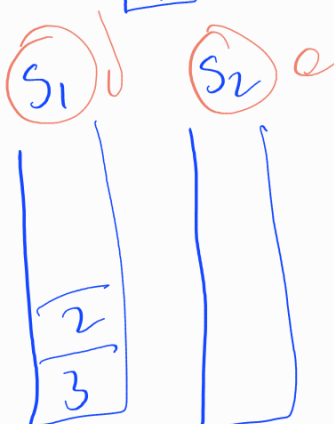
insert(3)



dequeue()

$O(1)$

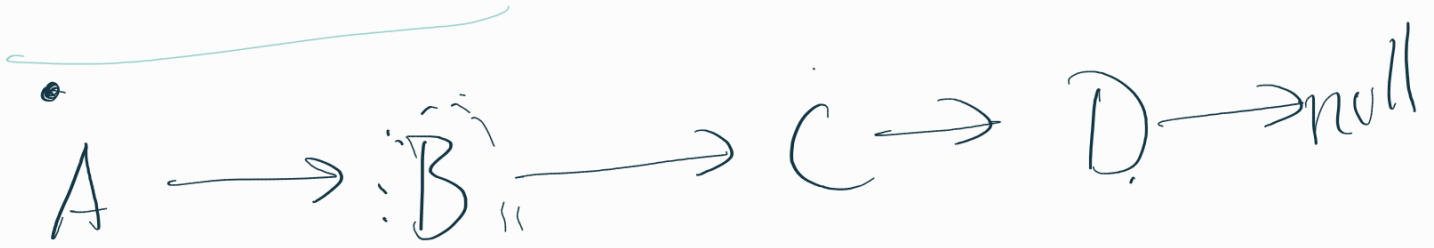
if d-stack not empty
if d-stack is empty



insert: check if dequeue stack is empty

dequeue: check if

degree is empty
after the operation



	before	after
A.next	B	null
B.next	C	A
C.next	D	B
D.next	null	C
head	A	D

$O(1)$ memory. temp.
 head?
 runtime $O(n)$

Diagram showing the reversed linked list structure:

```

null ← A ← B ← C ← D
  
```

Recursion - reverse str.

$S = 'ABCDE'$
 \downarrow
 reverse_str(s)
 \downarrow
 temp = 'A'
 $S_1 = 'BCDE'$
 \downarrow
 reverse_str(S₁)

recursion
returns

recursion
calls

\downarrow temp = 'B'
 $S_2 = \text{'CDE'}$
 $r_s(S_2)$
 \downarrow temp = 'C'
 $S_3 = \text{'DE'}$
 $r_s(S_3)$
 \downarrow temp = 'D'
 $S_4 = \text{'E'}$
 $r_s(S_4)$
 \downarrow temp = 'E'
 $S_5 = \text{''}$
 $r_s(S_5)$
return ""

" + 'E' + 'D' + 'C' + 'B' + 'A'
= 'EDCBA'

