SCHOOL OF
ENGINEERING
BANGKOK UNIVERSITY

Stack and Queue in Python

AIE 311 : Data structure and Algorithm

What is stack?

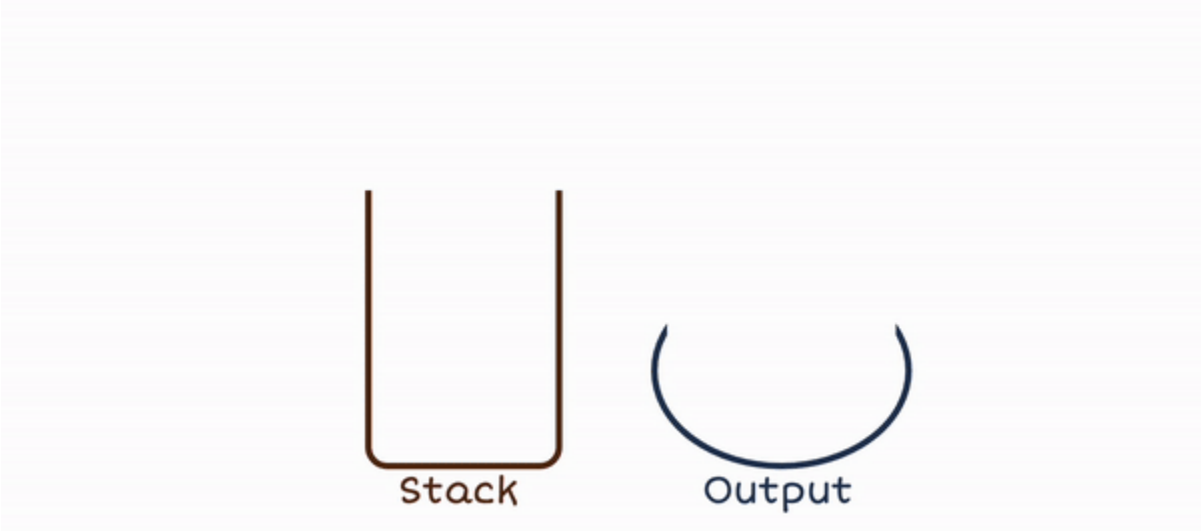
SCHOOL OF ENGINEERING
BANGKOK UNIVERSITY

Stack (Linear data structure)

- รูปแบบการจัดเก็บข้อมูล: Last In First Out (LIFO)

Stack Operation

- การเพิ่ม data ไว้ส่วนบนสุดของ stack (push)
- การลบ data ส่วนบนสุดออกจาก stack (pop)
- การดู data ส่วนบนสุดของ stack (peek)



Ref: https://www.borntodev.com/2023/09/26/stack-%E0%B9%81%E0%B8%A5%E0%B8%B0-queue/

Accessing stack (Push)

SCHOOL OF ENGINEERING BANGKOK UNIVERSITY

- Append the new value into the array (add new value at the last address)
  - Array.append() E.g., Array.append(8)

Address	0	1	2	3
Value	10	19	7	8
Direction	Bottom		Top	New Top

Array view

Address	Value	Direction
3	8	New Top
2	7	Top
1	19	
0	10	Bottom

Stack view

Accessing stack (Priority insert)

SCHOOL OF ENGINEERING BANGKOK UNIVERSITY

- Add the new value into the array (add new value depending on priority)

Address	0	1	2	3
Value	10	19	7	8
Priority	0	0	1	1
Direction	Bottom			Top

Before add

Address	0	1	2	3	4
Value	10	19	9	7	8
Priority	0	0	0	1	1
Direction	Bottom				Top

Added where priority = 0

Accessing stack (Pop)

- Remove the last value from the queue (pop from top / above)
  - Array.pop()

Address	0	1	2	3
Value	10	19	7	8
Direction	Bottom		New Top	Top

Array view

Address	Value	Direction
3	8	Top
2	7	New Top
1	19	
0	10	Bottom

Stack view

Accessing stack (Pop) cont.

- Remove the first value from the queue (pop from bottom)
  - Array.pop(0)

Address	0	1 -> 0	2 -> 1	3 -> 2
Value	10	19	7	8
Direction	Bottom	New B.		Top

Array view

Address	Value	Direction
3 -> 2	8	Top
2 -> 1	7	
1 -> 0	19	New B.
0	10	Bottom

Stack view

Accessing stack (Peak)

SCHOOL OF ENGINEERING BANGKOK UNIVERSITY

- Peak from bottom
  - Bottom = Array[0]

Address	0	1	2	3
Value	10	19	7	8
Direction	Bottom			Top

Array view

Address	Value	Direction
3	8	Top
2	7	
1	19	
0	10	Bottom

Stack view

Accessing stack (Peak) cont.

SCHOOL OF ENGINEERING BANGKOK UNIVERSITY

- Peak from top
  - Bottom = Array[-1]

Address	0	1	2	3
Value	10	19	7	8
Direction	Bottom			Top

Array view

Address	Value	Direction
3	8	Top
2	7	
1	19	
0	10	Bottom

Stack view

Accessing stack (Queue length)

- Length = len(Array)

Address	0	1	2	3
Value	10	19	7	8
Direction	Bottom			Top

Array view

Address	Value	Direction
3	8	Top
2	7	
1	19	
0	10	Bottom

Stack view

Queue data flow

SCHOOL OF ENGINEERING BANGKOK UNIVERSITY

Queue (Linear data structure)

- รูปแบบการจัดเก็บข้อมูล: First In First Out (FIFO)

Queue Operation

- การเพิ่ม data ไว้ส่วนบนสุดของ queue (enqueuer())
- การลบ data ส่วนบนสุดออกจาก queue (dequeuer())
- การดู data ตำแหน่งท้ายสุด (rear())

Ref: https://www.borntodev.com/2023/09/26/stack-%E0%B9%81%E0%B8%A5%E0%B8%B0-queue/

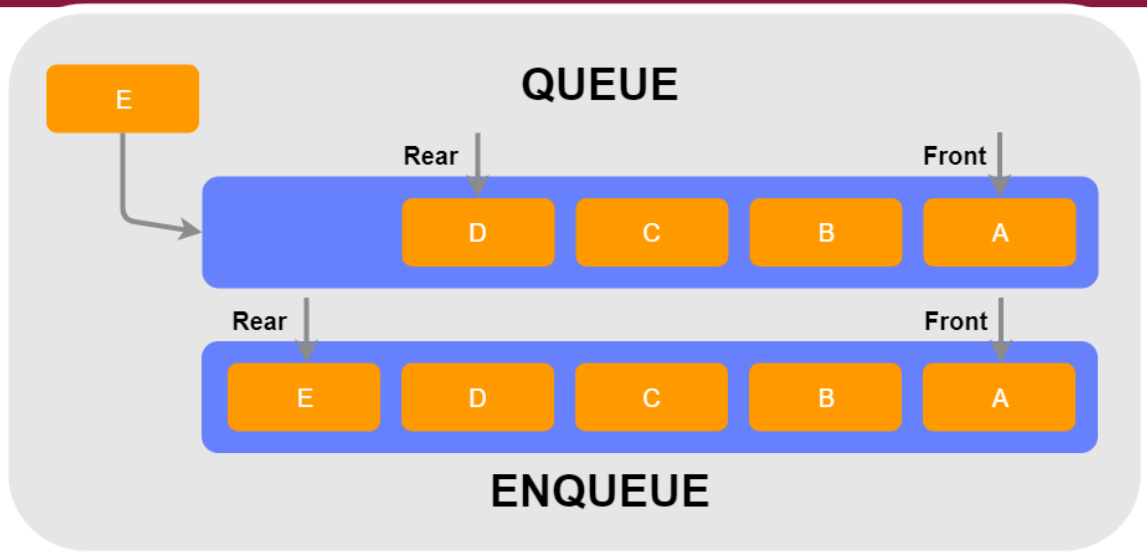
Queue data flow

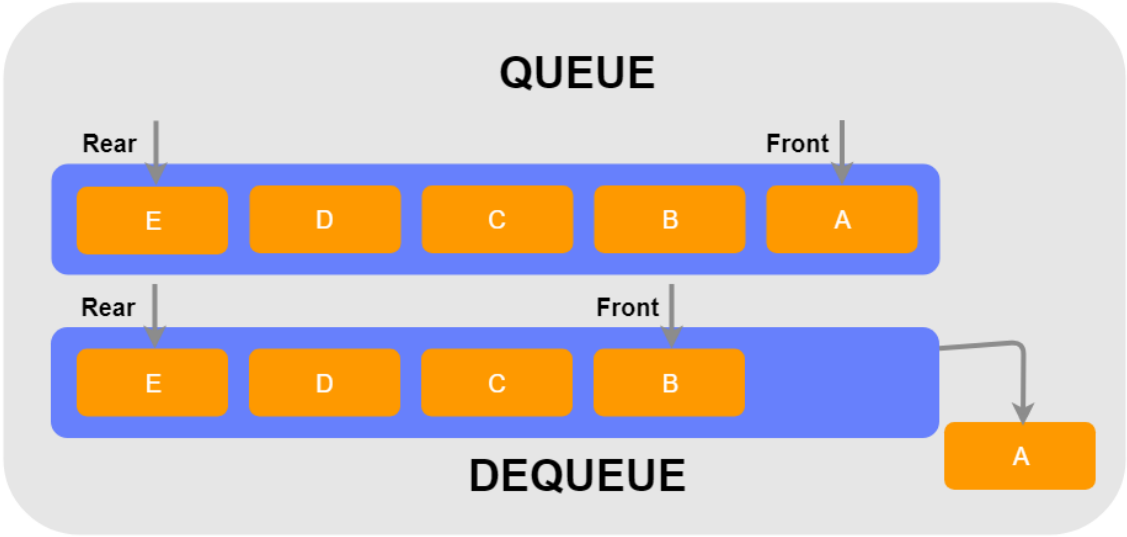
Queue (Linear data structure)

- รูปแบบการจัดเก็บข้อมูล: First In First Out (FIFO)

Queue Operation

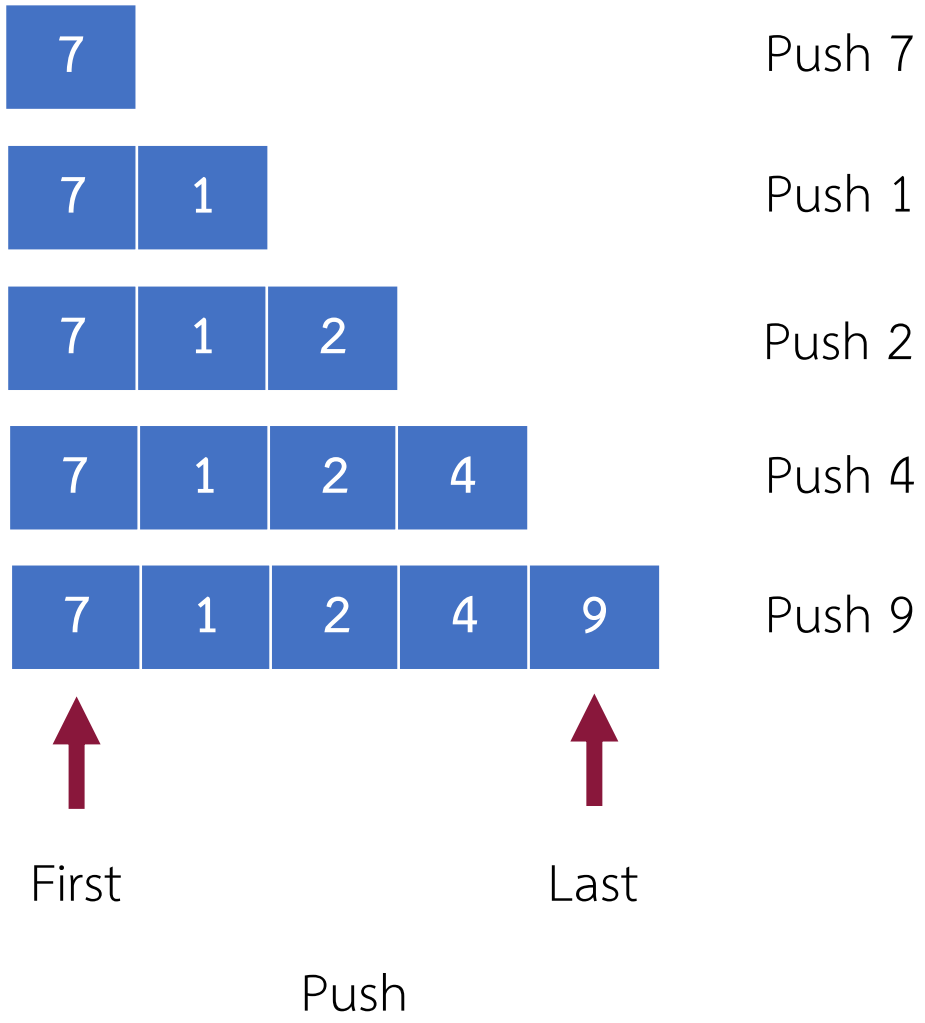
- การเพิ่ม data ไว้ส่วนบนสุดของ queue (enqueuer())
- การลบ data ส่วนบนสุดออกจาก queue (dequeuer())
- การดู data ตำแหน่งท้ายสุด (rear())

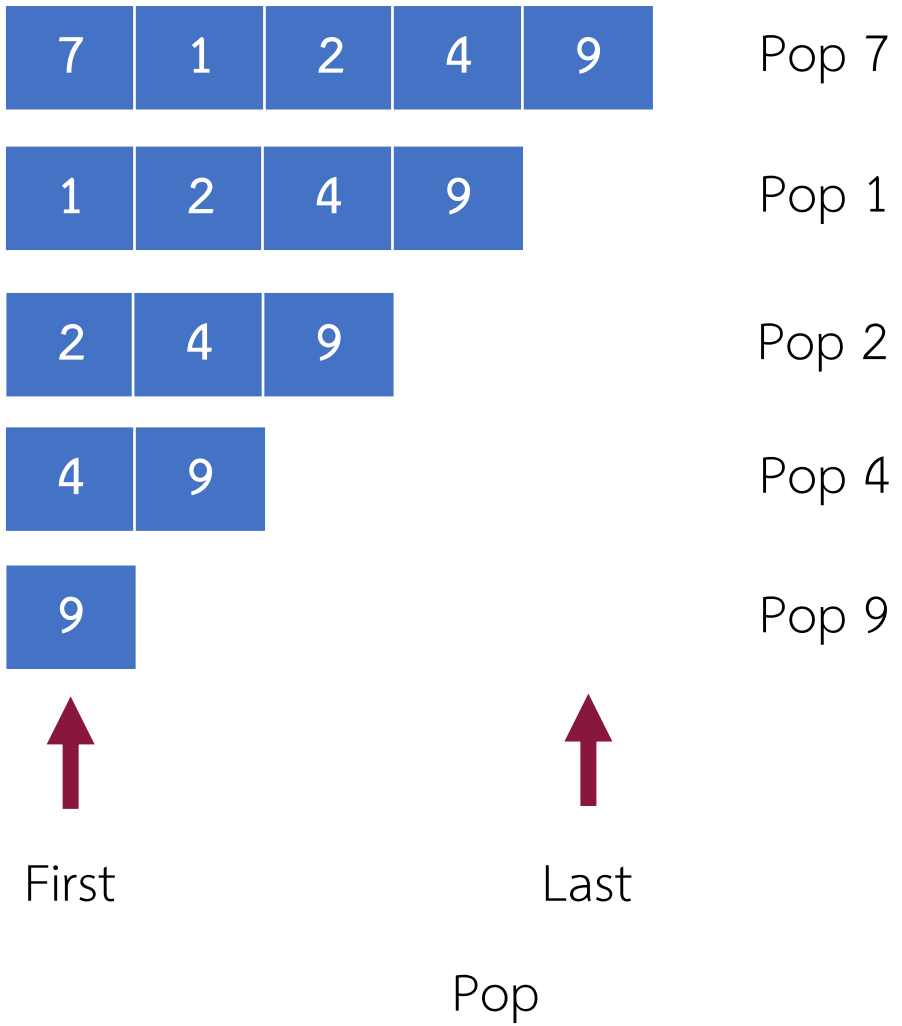




Ref: https://shorturl.asia/uh1jc

Queue data flow (FIFO)





In, Pre, Postfix

SCHOOL OF
ENGINEERING
BANGKOK UNIVERSITY

- Stack, Queue can use to reverse array
- In this case stack can use to transform Infix into Prefix and Postfix
  - Infix = A + B
  - Prefix = +AB
  - Postfix = AB+

Infix to Prefix

- Example: Conversion to A + B - C

Operator	Output
	A
+	A
+	AB
	+AB
-	+AB
-	+ABC
	-+ABC

Infix to Postfix

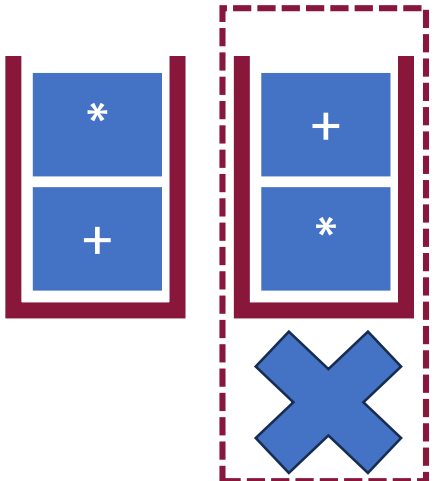
- Example: Conversion to A + B - C

Operator	Output
	A
+	A
+	AB
	AB+
-	AB+
-	AB+C
	AB+C-

Operand priority (Short)

Priority	Operator
Highest (0)	( )
(1)	^
(2)	* /
Lowest (3)	+ -

- Lower priority cannot stack over higher priority



Example	
+*	True : * has higher priority than +
*+	False : + has lower priority than -

Infix to Prefix (with priority)

- Example: Conversion to A + (B – C)

Operator	Output
	A
+	A
+(	A
+(	AB
+(-	AB
+(-	ABC
+	ABC-
	ABC-+