Review

- Arrays
  - A sequence of neighboring memory boxes
    - Know where an **arbitrary** (**i-th**) element is located, by using the **neighboring rule**
  - **Limitation**: Fixed length and Expensive resizing
    - Make a brand-new array + copy all the existing elements
  - **Improvement**: Resizing step adjustment
- Linked lists
  - A list of nodes each of which has a link to another node
    - Know where the **next** element is located, by using the **next pointer**
  - **Limitation:** Don't know what is where - Frequent navigation through the list
  - **Improvement:** Caching and sentinel

Queues

Lecture 14-1

Hyung-Sin Kim


SNU Graduate School of Data Science

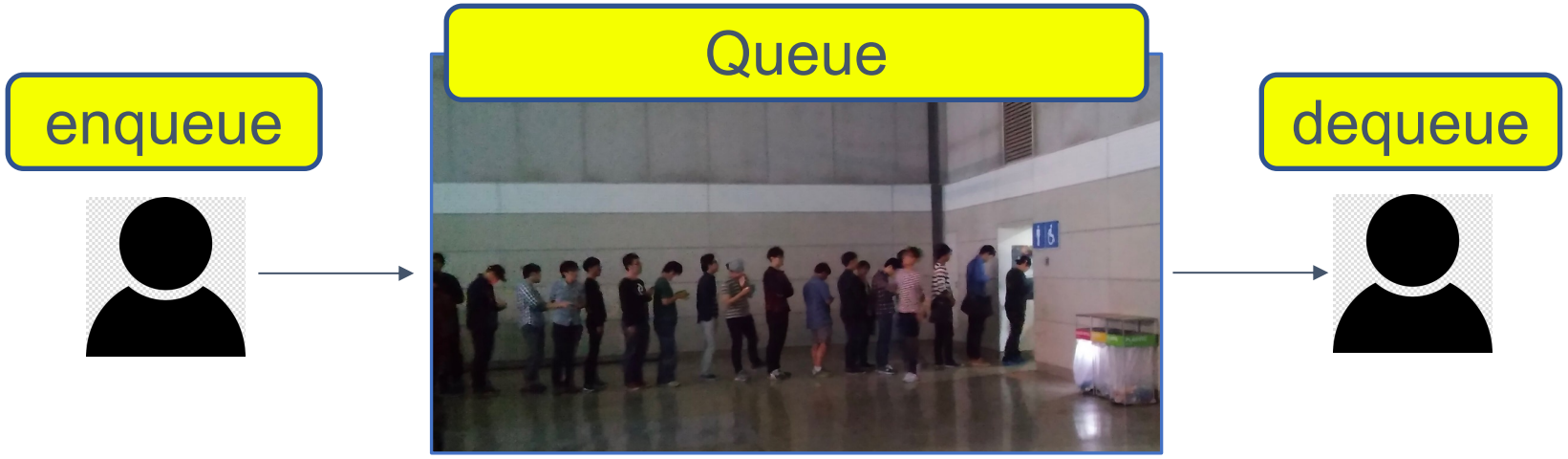
Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

enqueue()

dequeue()

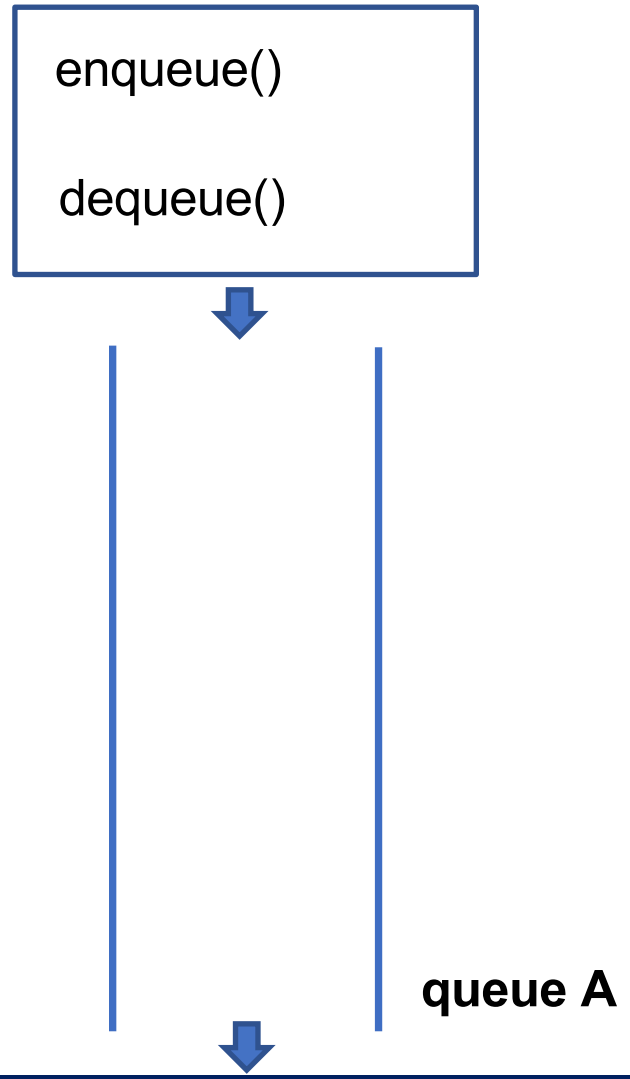
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element from the queue



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

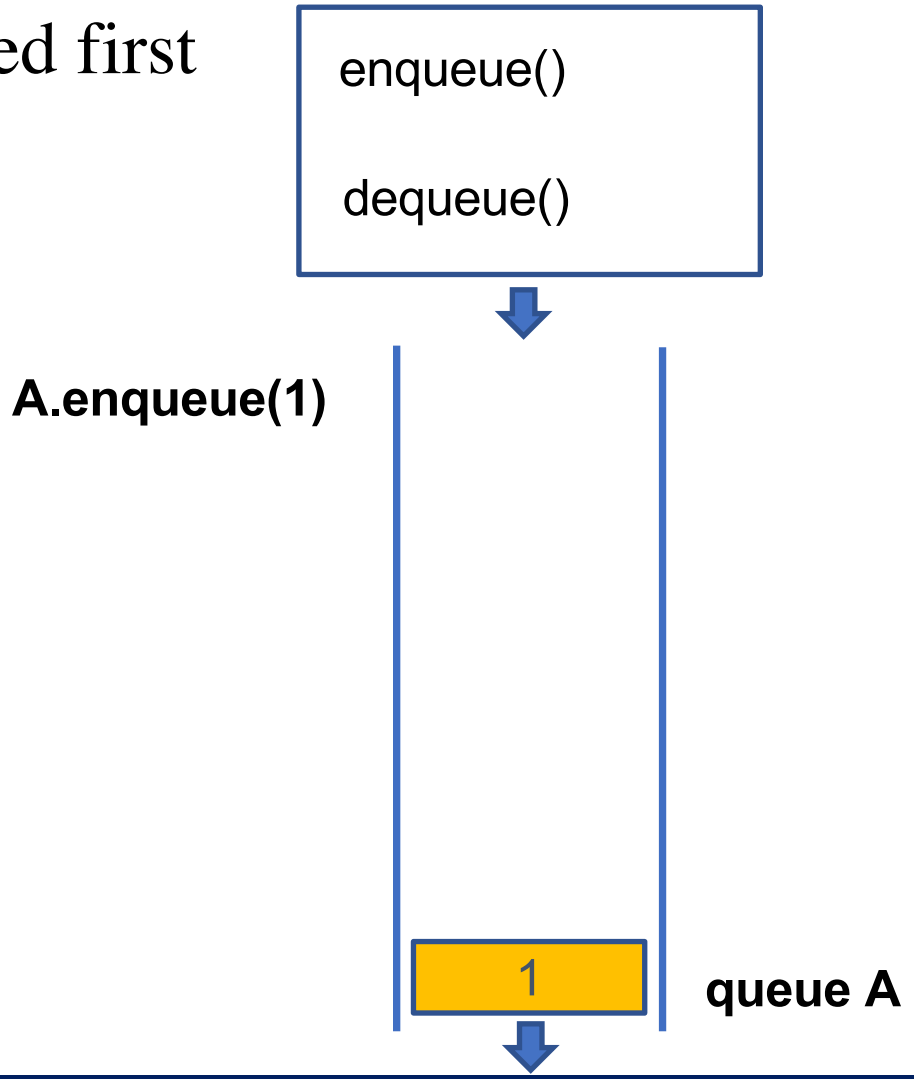
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

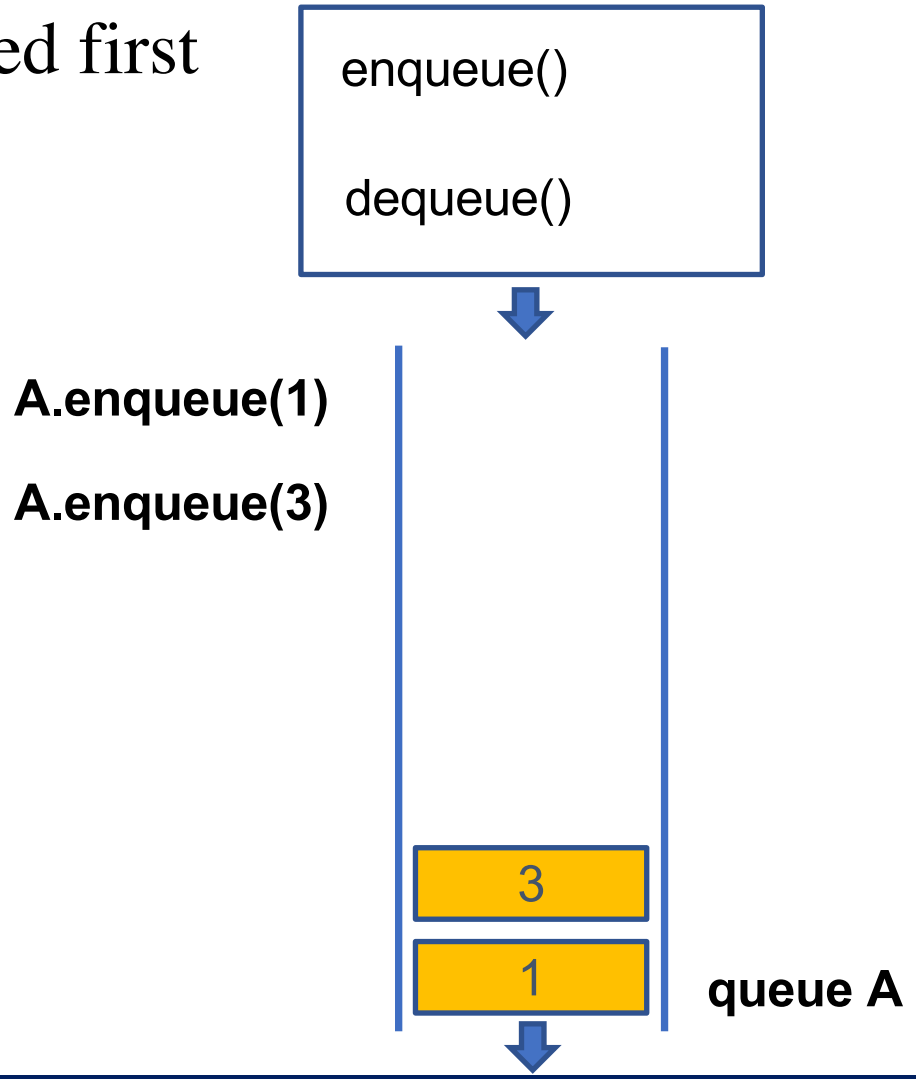
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

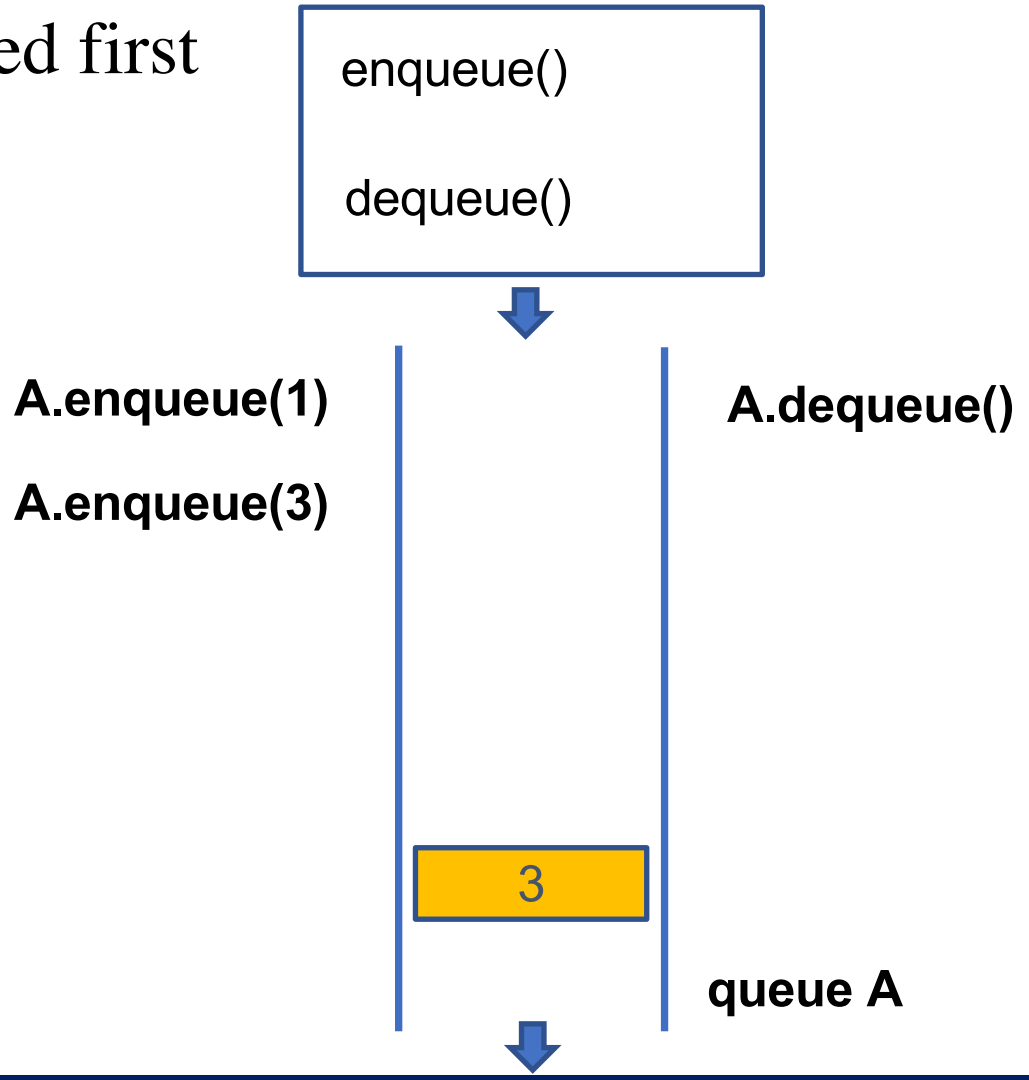
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

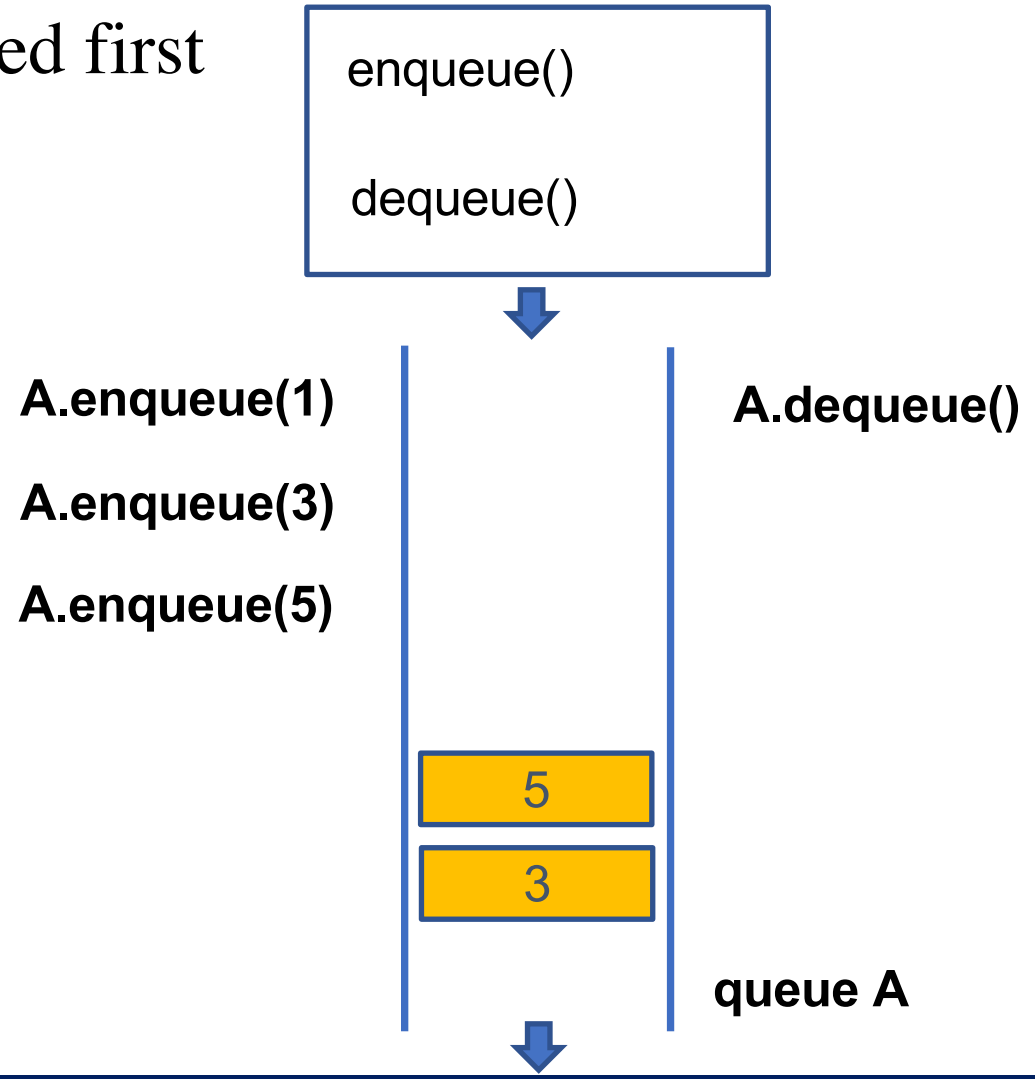
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

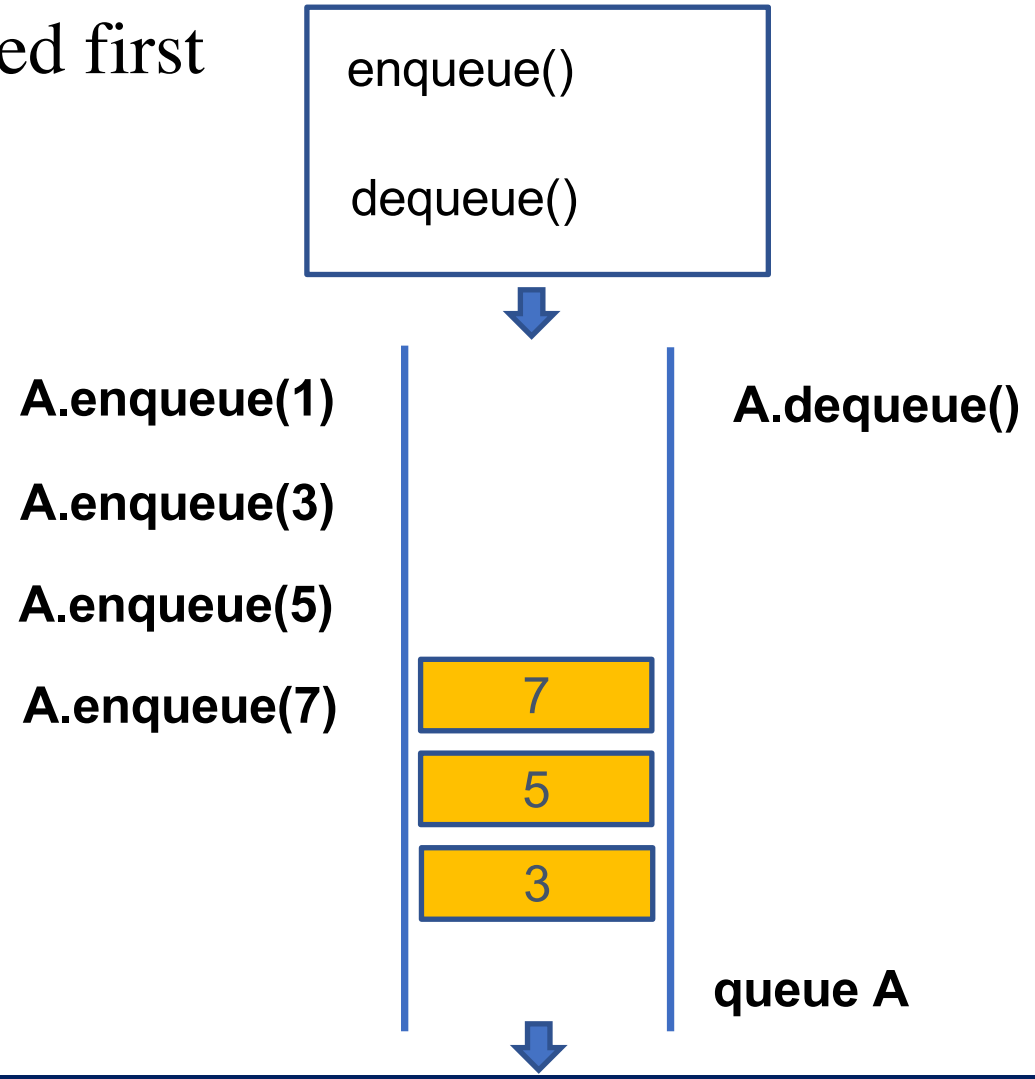
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

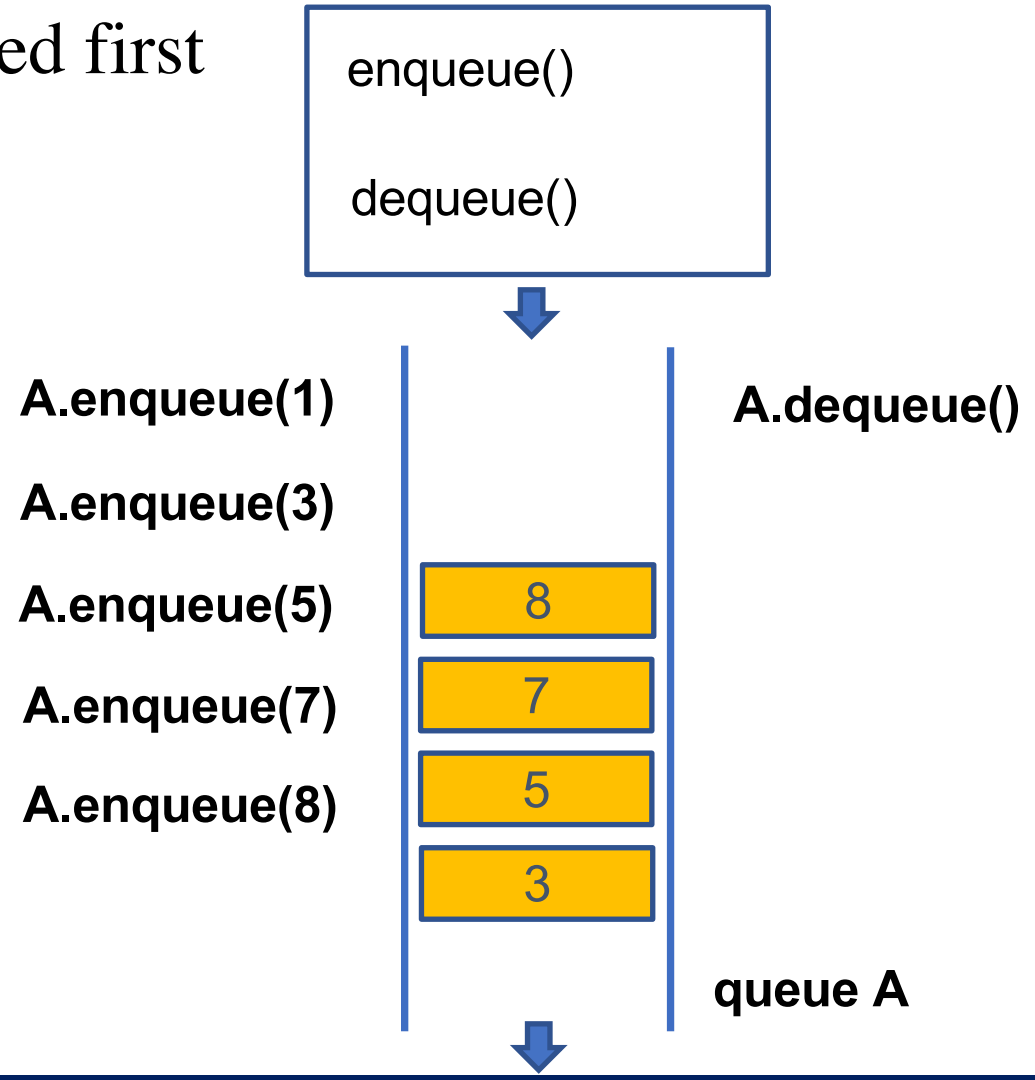
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

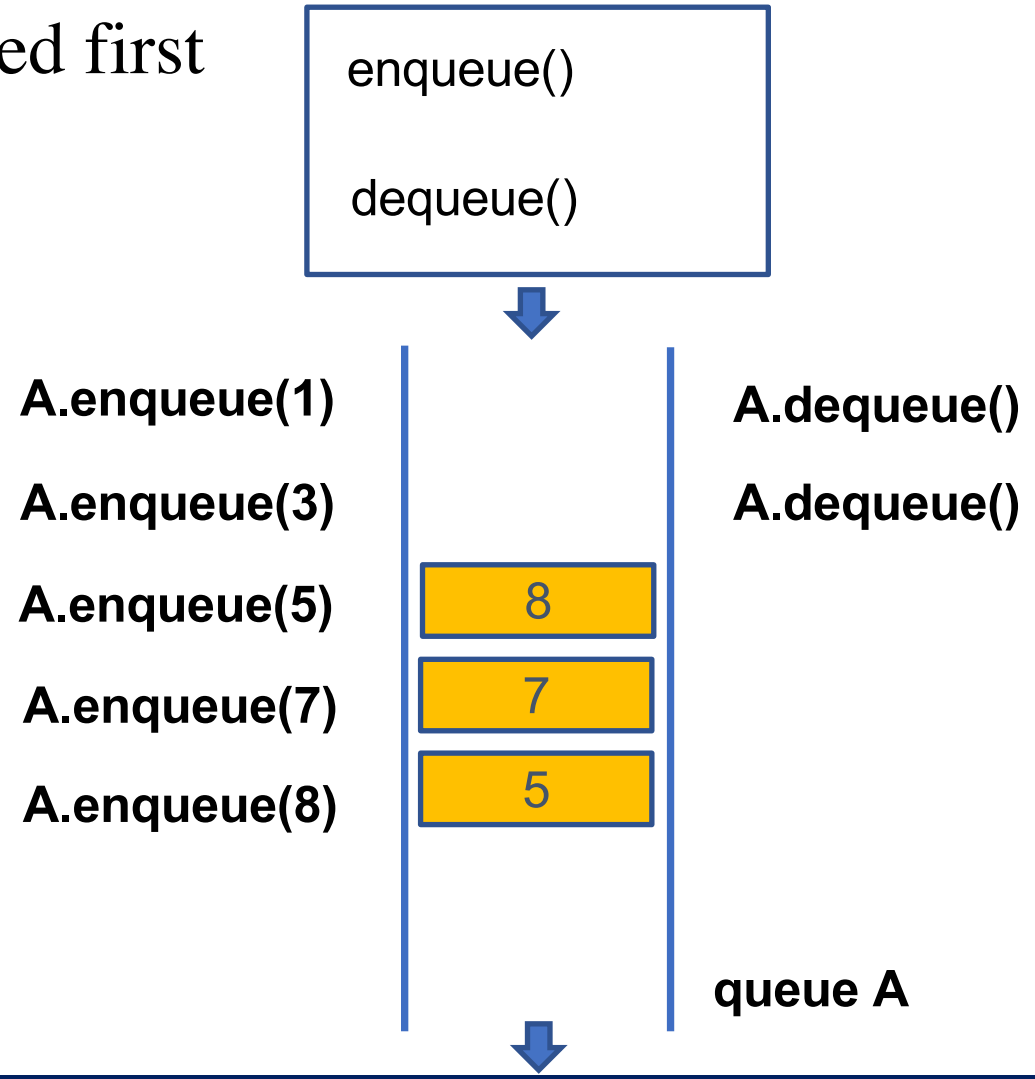
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

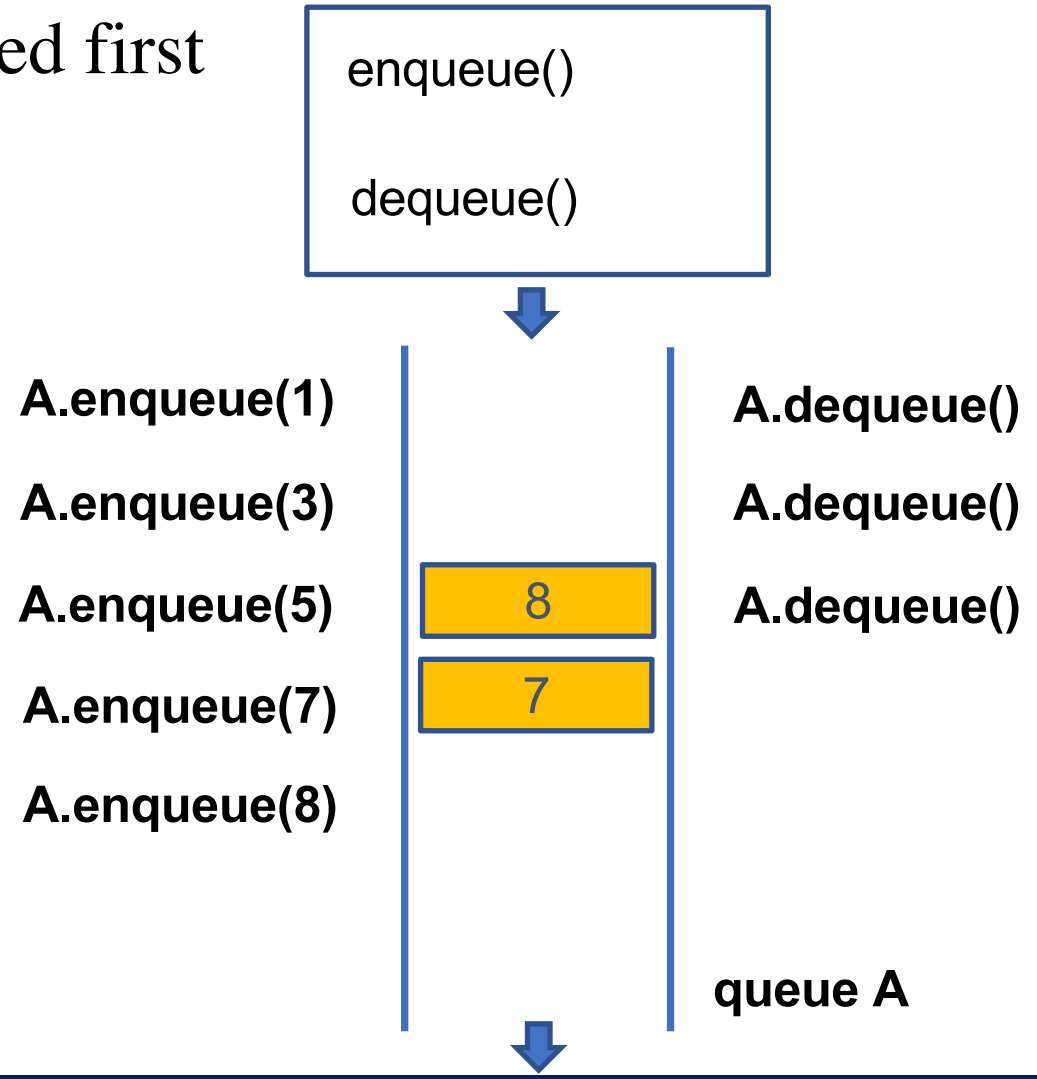
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

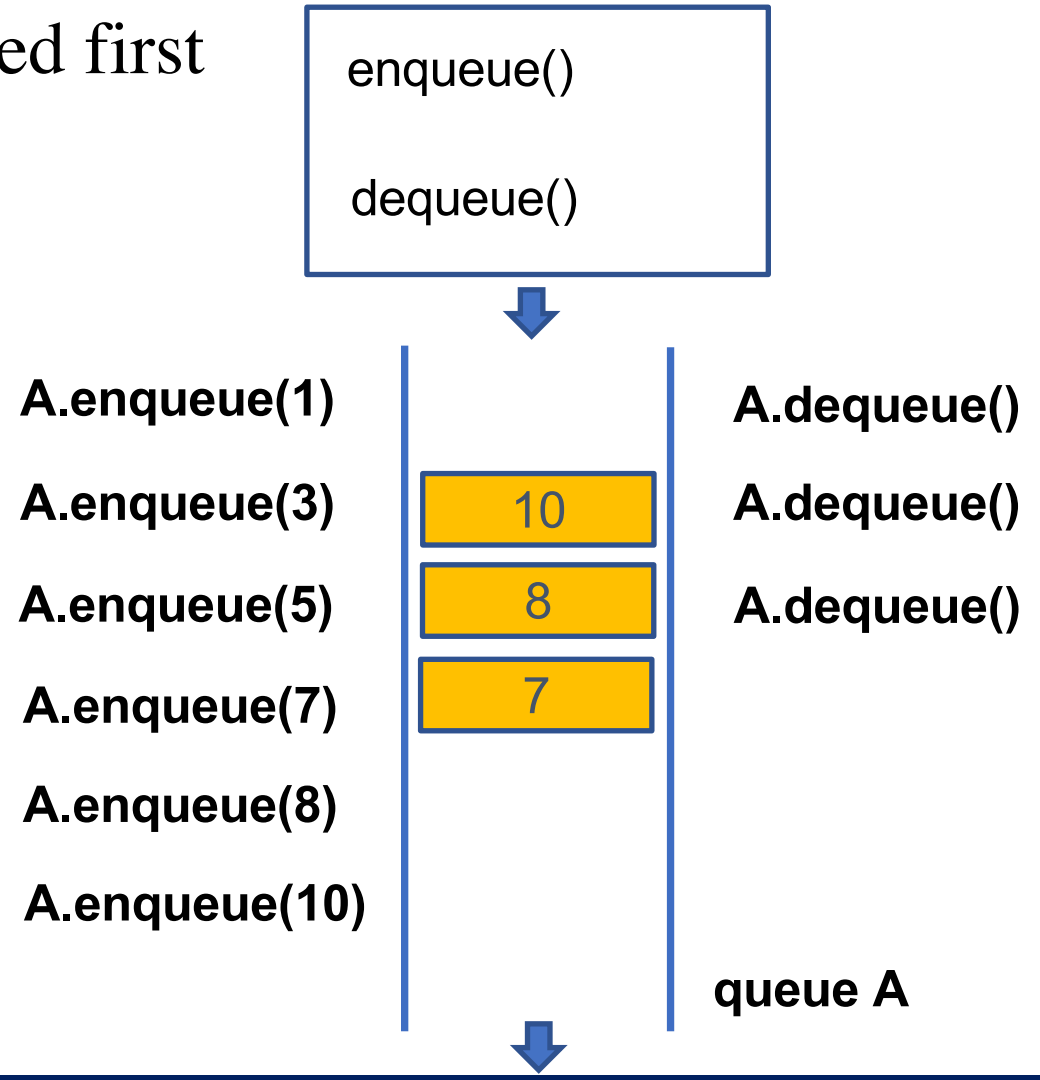
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

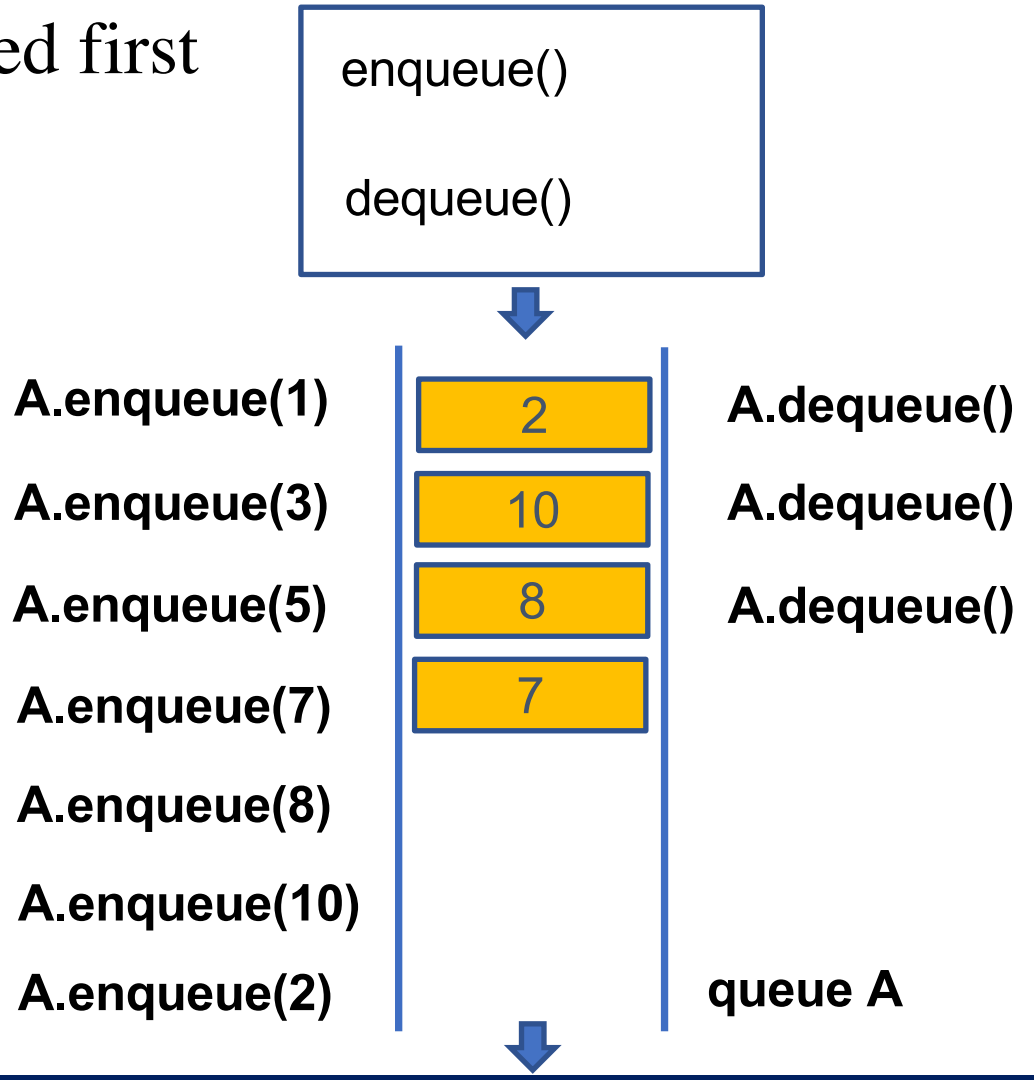
- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Queue – a First In and First Out Data Structure

- FIFO – First enqueued element is dequeued first

- Queue has two methods
  - enqueue(): add an element to the queue
  - dequeue(): remove the oldest element



Stacks

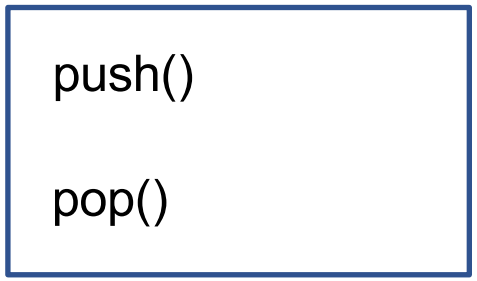
Lecture 14-2

Hyung-Sin Kim

SNU Graduate School of Data Science

Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

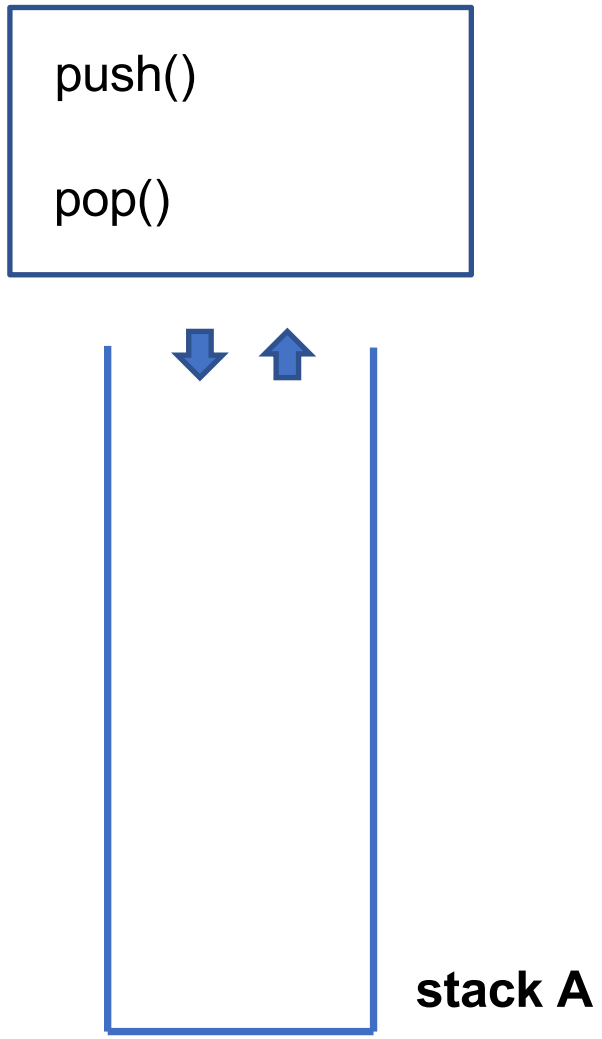


- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack

Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

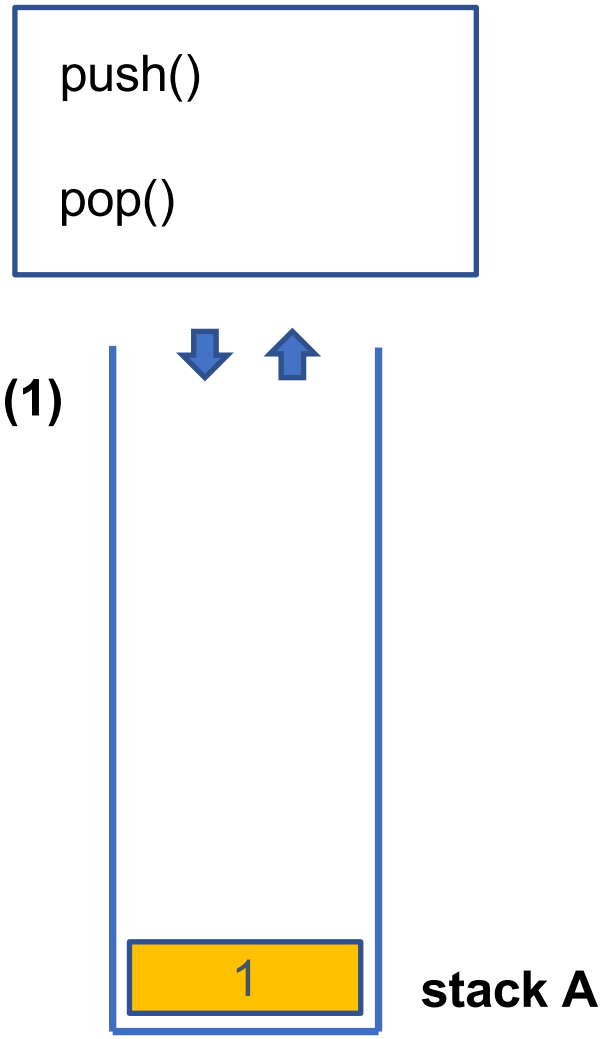
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

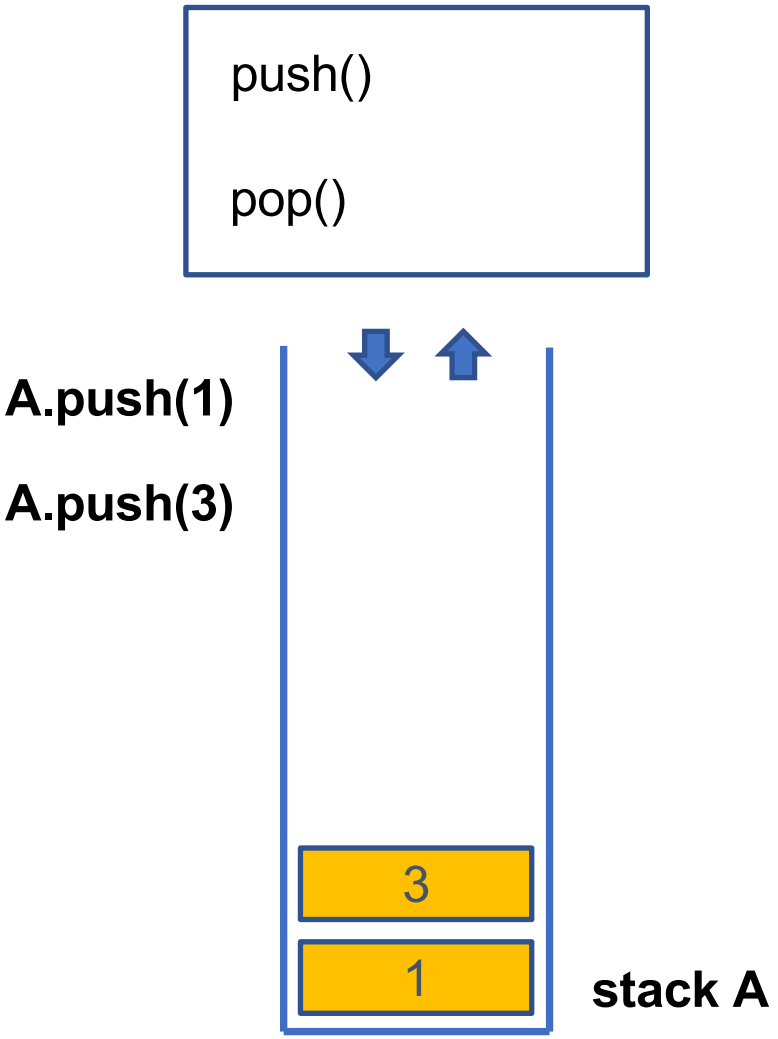
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

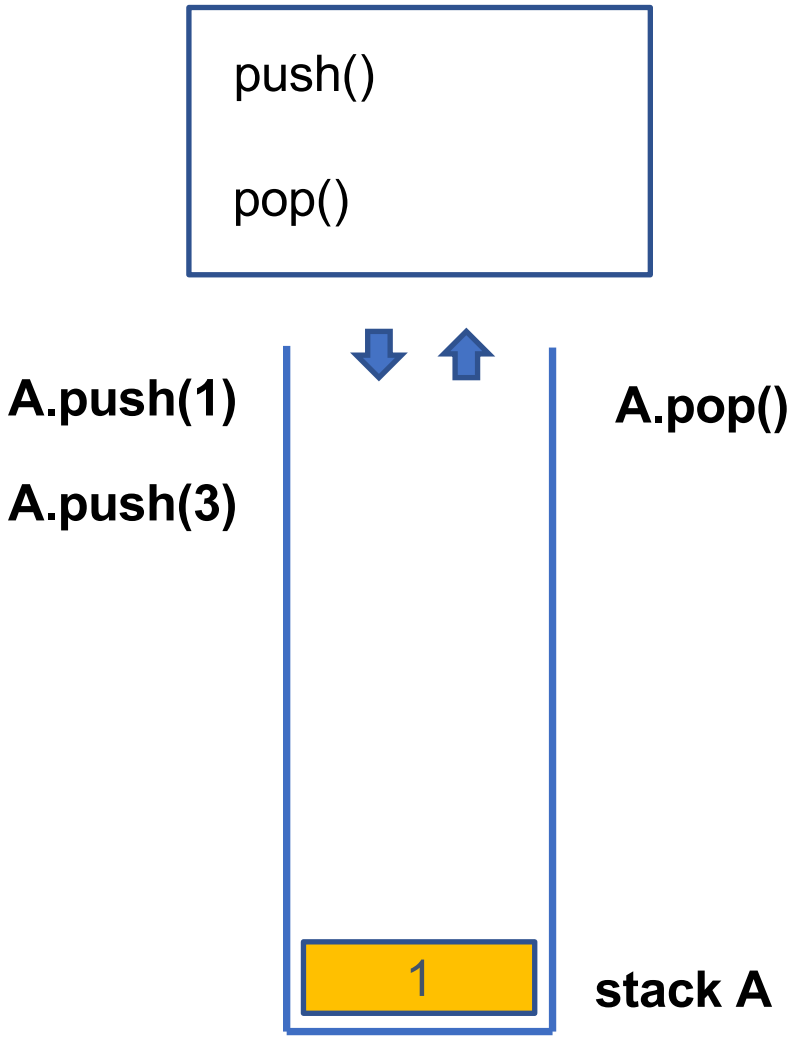
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

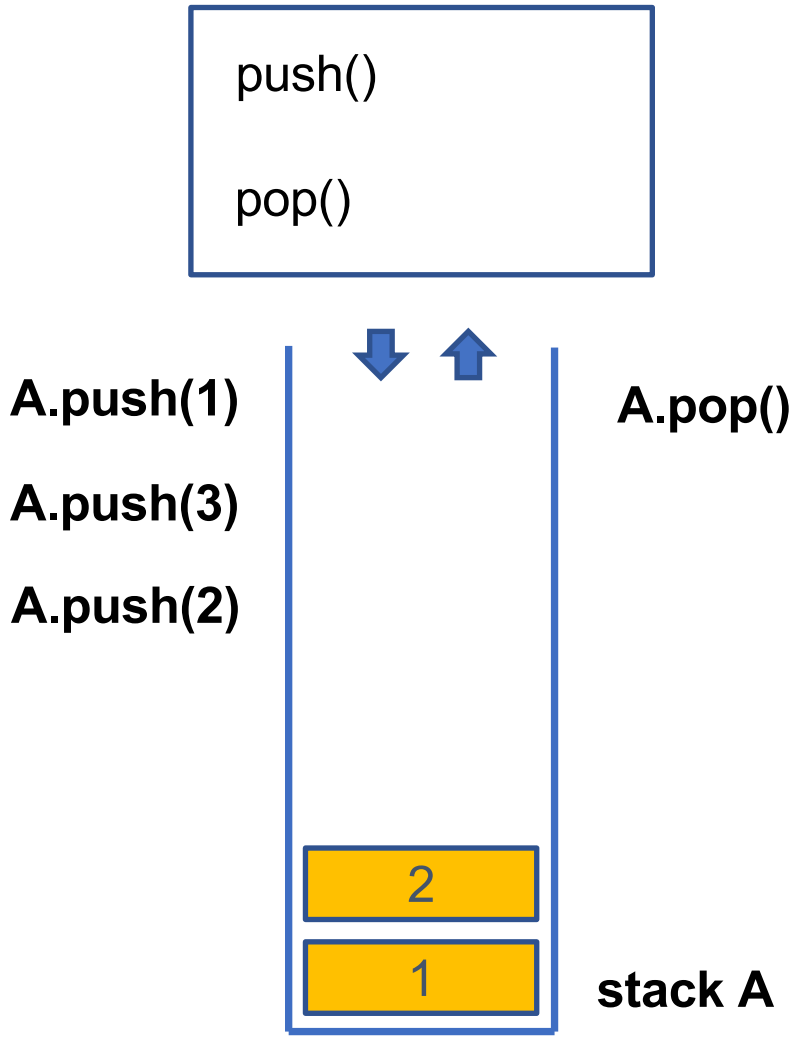
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

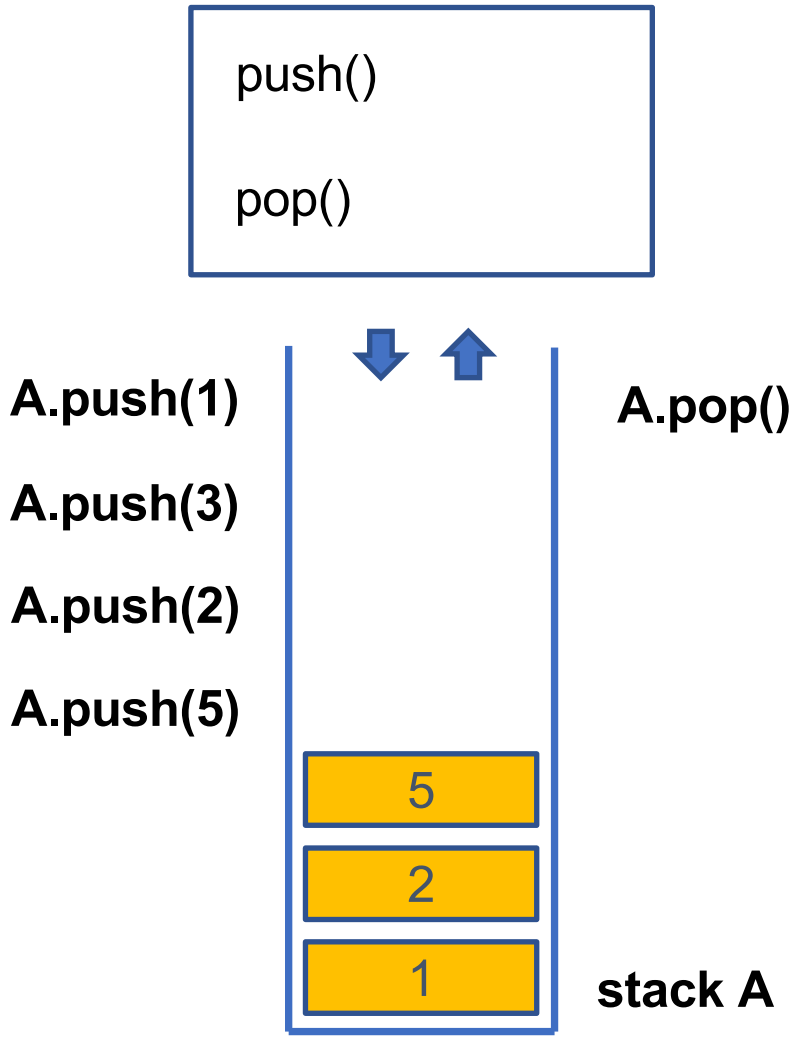
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

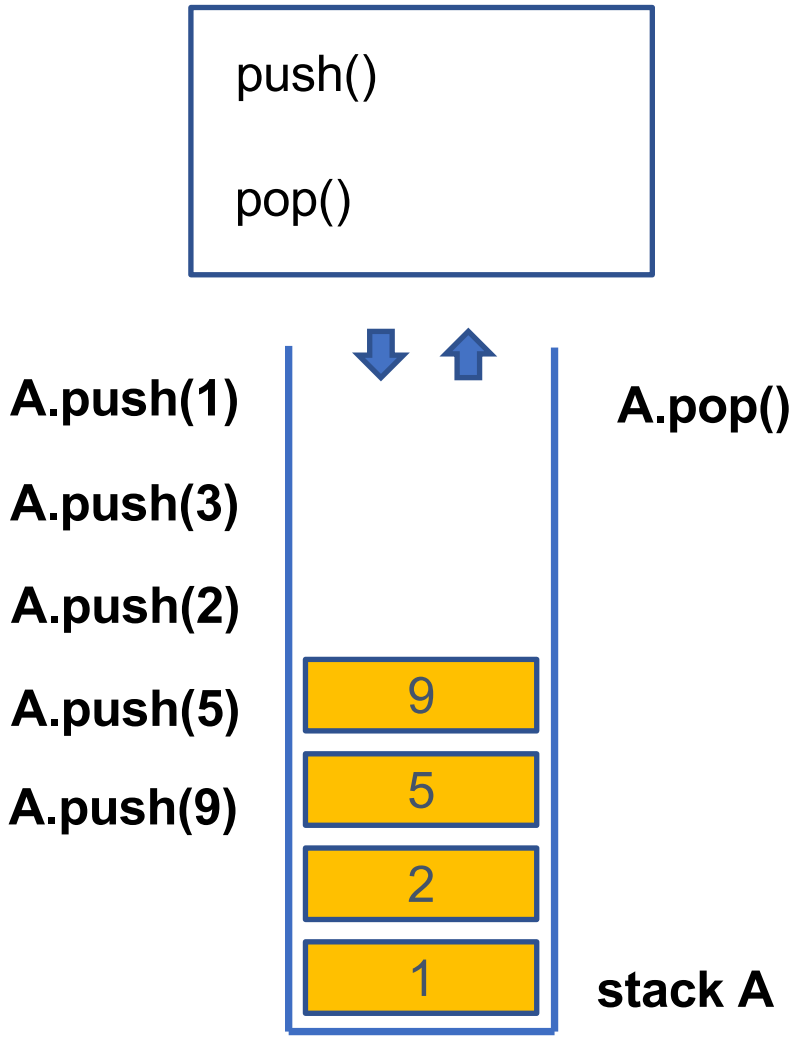
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

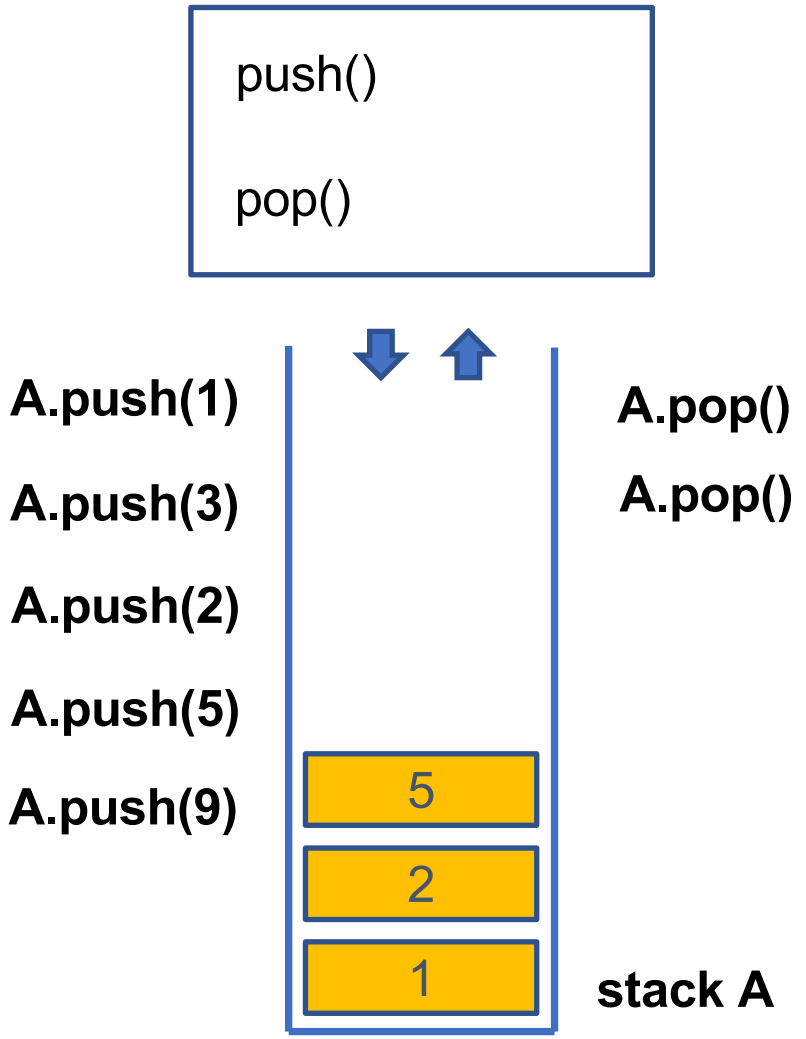
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

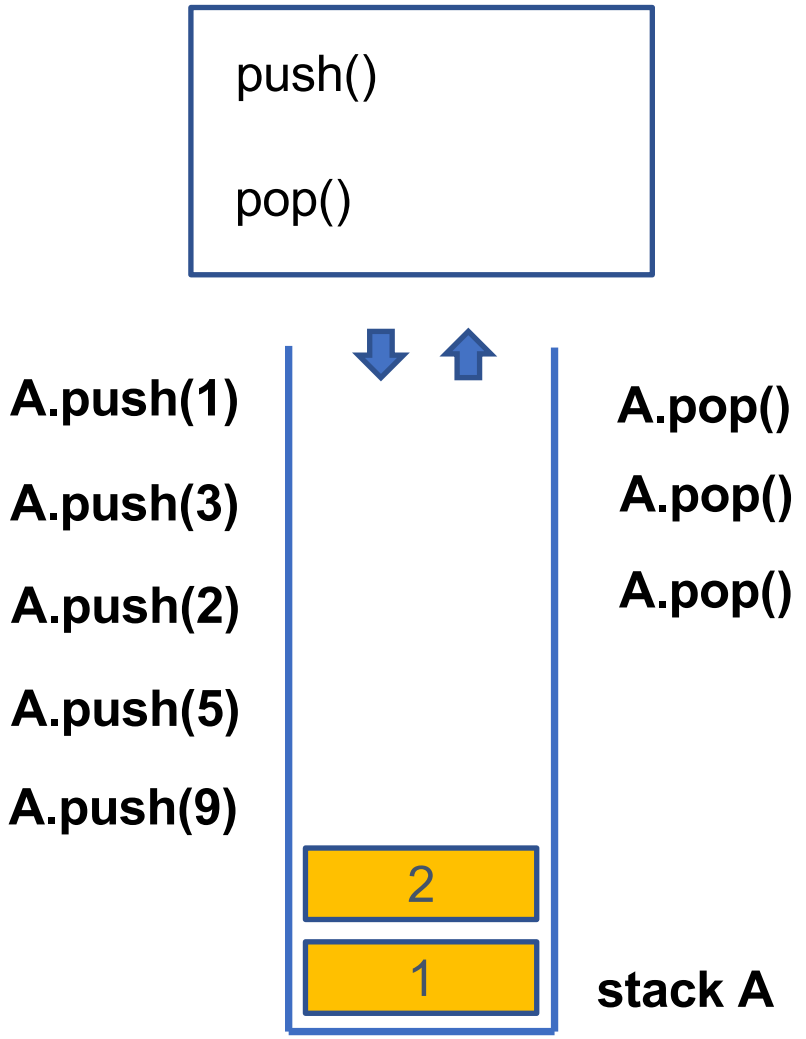
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

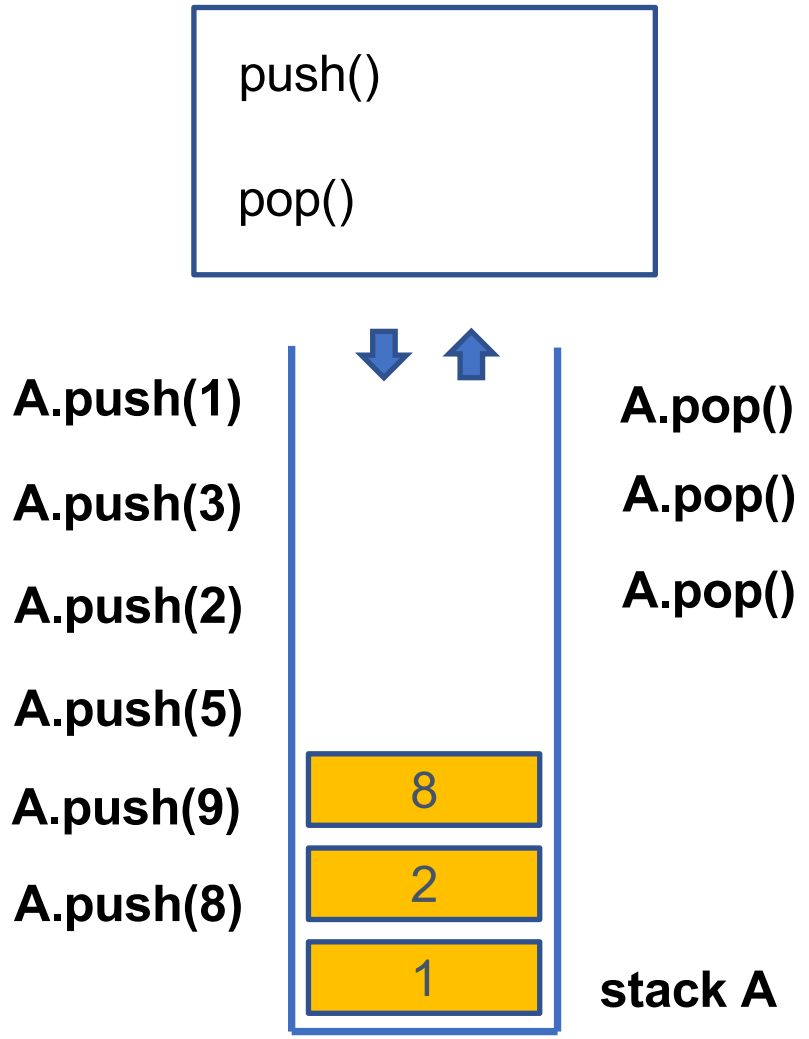
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

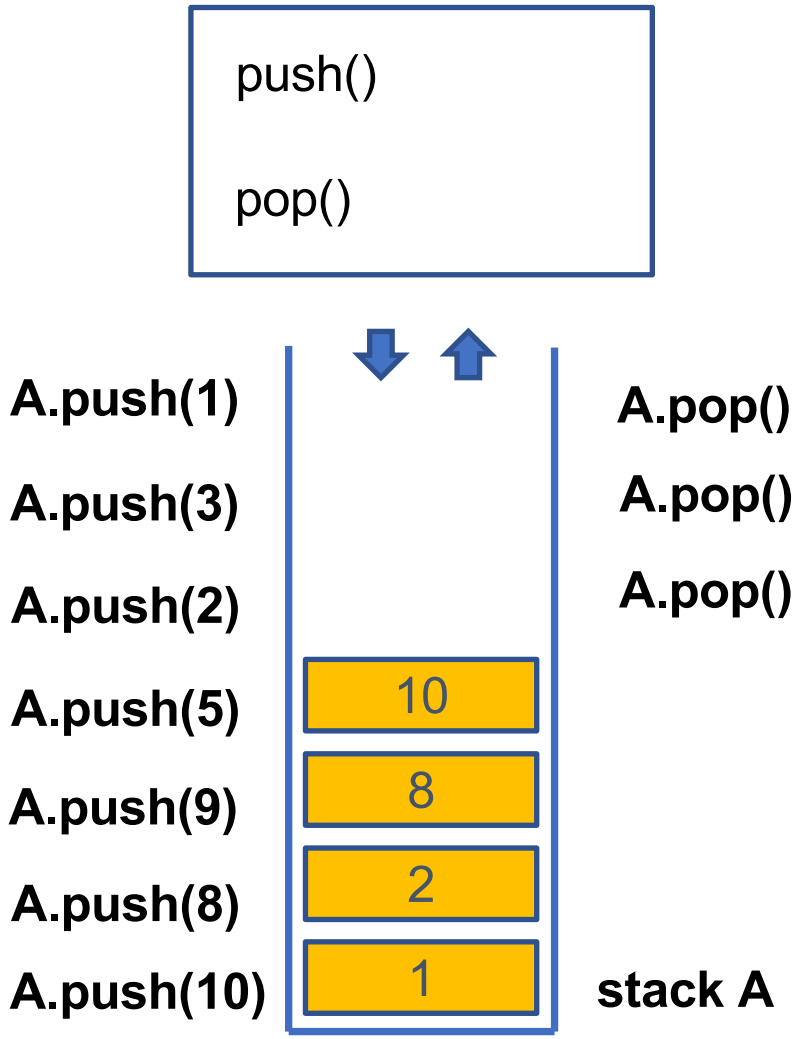
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first

- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack



Stack – a Last In and First Out Data Structure

- LIFO – Last pushed element is popped first
- Stack has two methods
  - push(): add an element to the stack
  - pop(): remove the newest element from the stack
- Use cases
  - Undo function: Ctrl + z
  - Parentheses matching: (){}[])