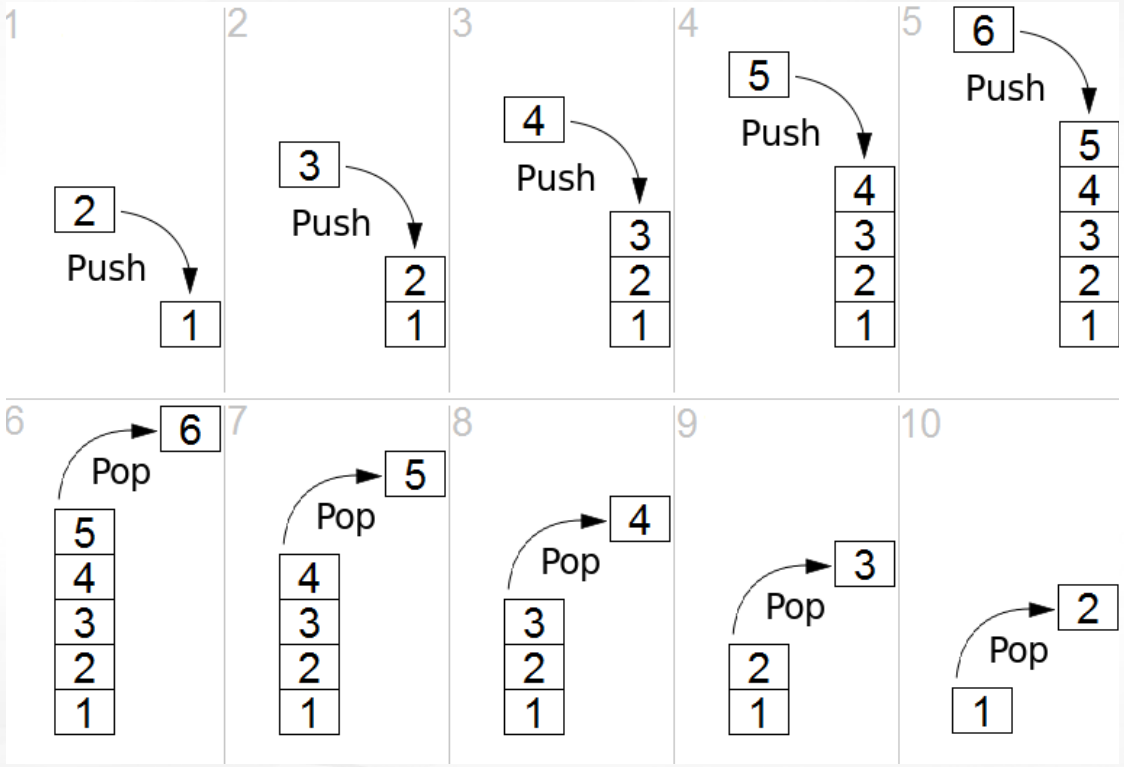
Udemy

Algorithms and Data Structures in Java

Lecture: Stacks



Instructors:
George Katsilidis
Nikos Katsilidis
Christos Topalidis

STACK LIFO (LAST IN - FIRST OUT)

Representing stacks with arrays is a natural idea. In particular, we maintain an instance variable `n` that stores the number of items in the stack and an array `items[]` that stores the `n` items, with the most recently inserted item in `items[n-1]` and the least recently inserted item in `items[0]`.

```
private int[] items;            //Array of integers
private int n;                  //Number of elements on stack

//Create an empty stack
public Stack(){
        items = new int[2];
}
```

```
private int[] items;                    //Array of integers
private int n;                  //Number of elements on stack

//Create an empty stack
public Stack(int capacity){
        items = new int[capacity];
}
```

STACK LIFO (LAST IN - FIRST OUT)
createStack(),isEmpty(),isFull()

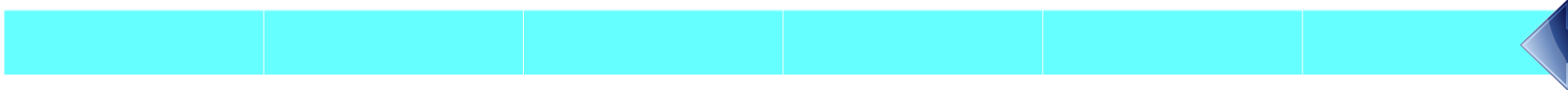
```
capacity = 6;
public createStack(int capacity) {
    items = new int[capacity];
}
```

STACK LIFO (LAST IN - FIRST OUT)
createStack(),isEmpty(),isFull()

```
capacity = 6;
public createStack(int capacity) {
    items = new int[capacity];
 }
```



```
public boolean isEmpty() {
    return n == 0;
}
```



Empty Stack

STACK LIFO (LAST IN - FIRST OUT)
createStack(),isEmpty(),isFull()

```
capacity = 6;
public createStack(int capacity) {
    items = new int[capacity];
 }
```

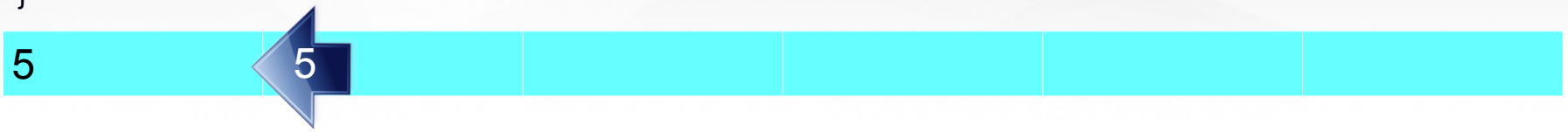


```
public boolean isEmpty() {
    return n == 0;
}
```


```
public boolean isFull() {
    return n == items.length;
}
```

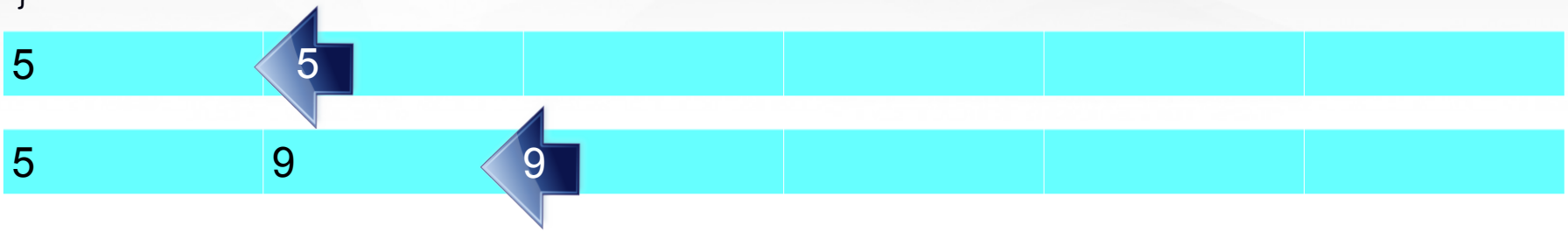
STACK LIFO (LAST IN - FIRST OUT)
push()

```
item = 5;
public void push(int item) {
    items[n++] = item;
}
```



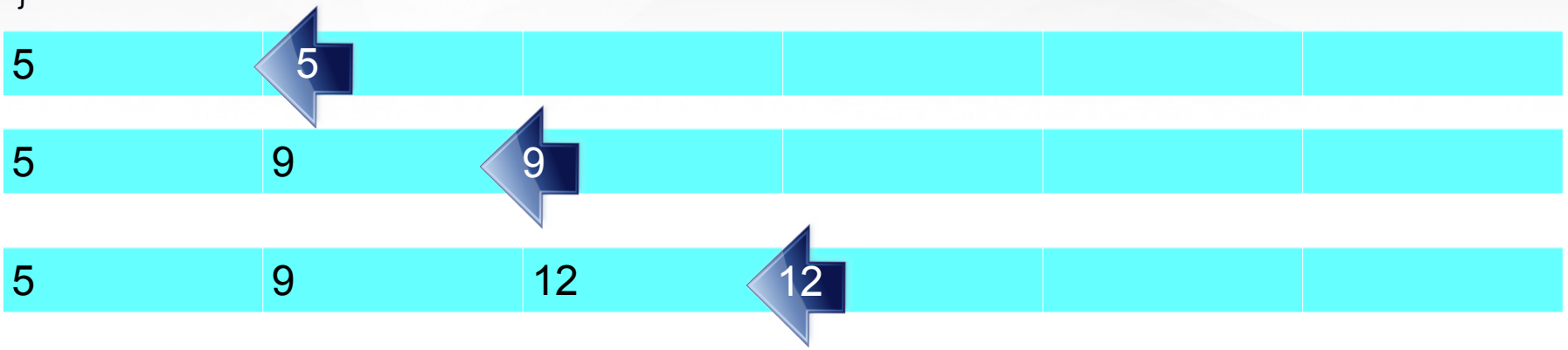
STACK LIFO (LAST IN - FIRST OUT)
push()

```
item = 9;
public void push(int item) {
    items[n++] = item;
}
```



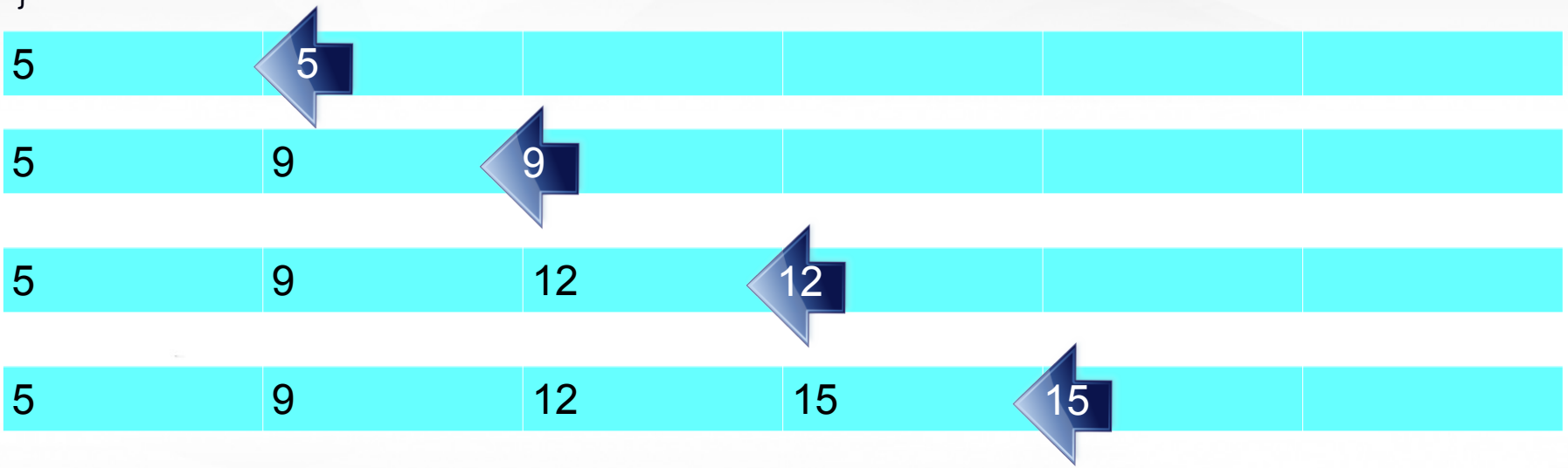
STACK LIFO (LAST IN - FIRST OUT)
push()

```
item = 12;
public void push(int item) {
    items[n++] = item;
}
```



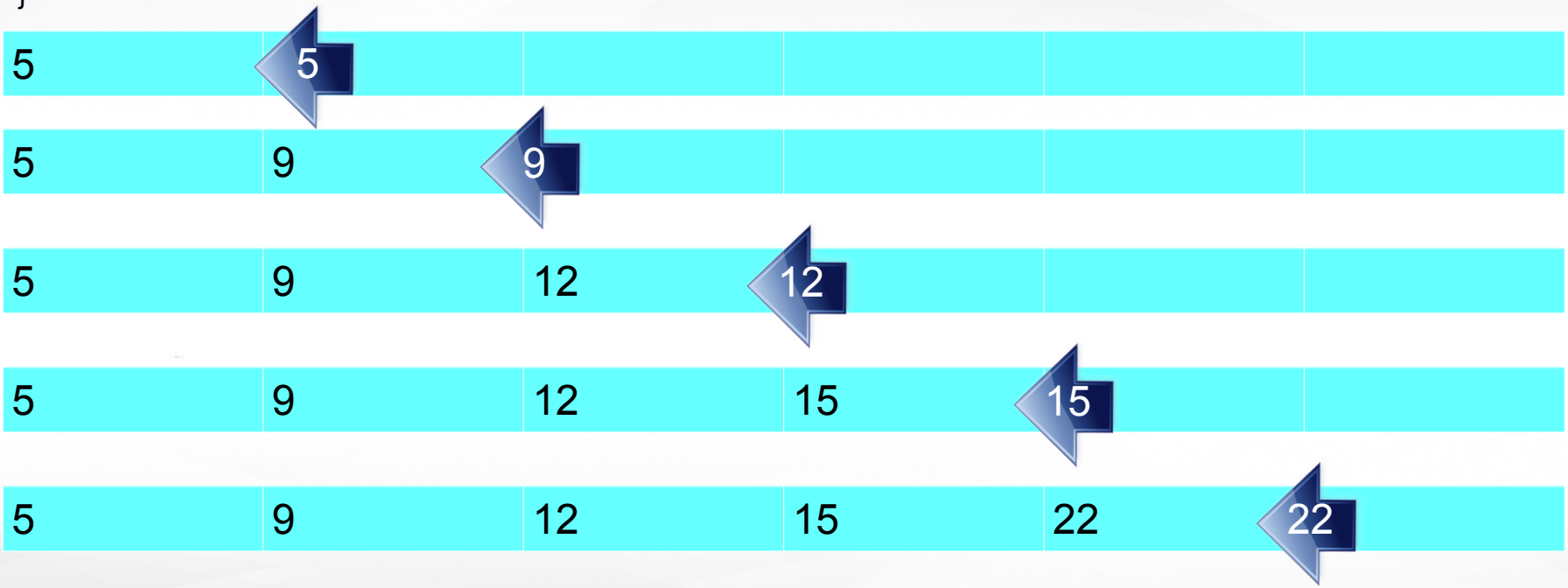
STACK LIFO (LAST IN - FIRST OUT)
push()

```
item = 15;
public void push(int item) {
    items[n++] = item;
}
```



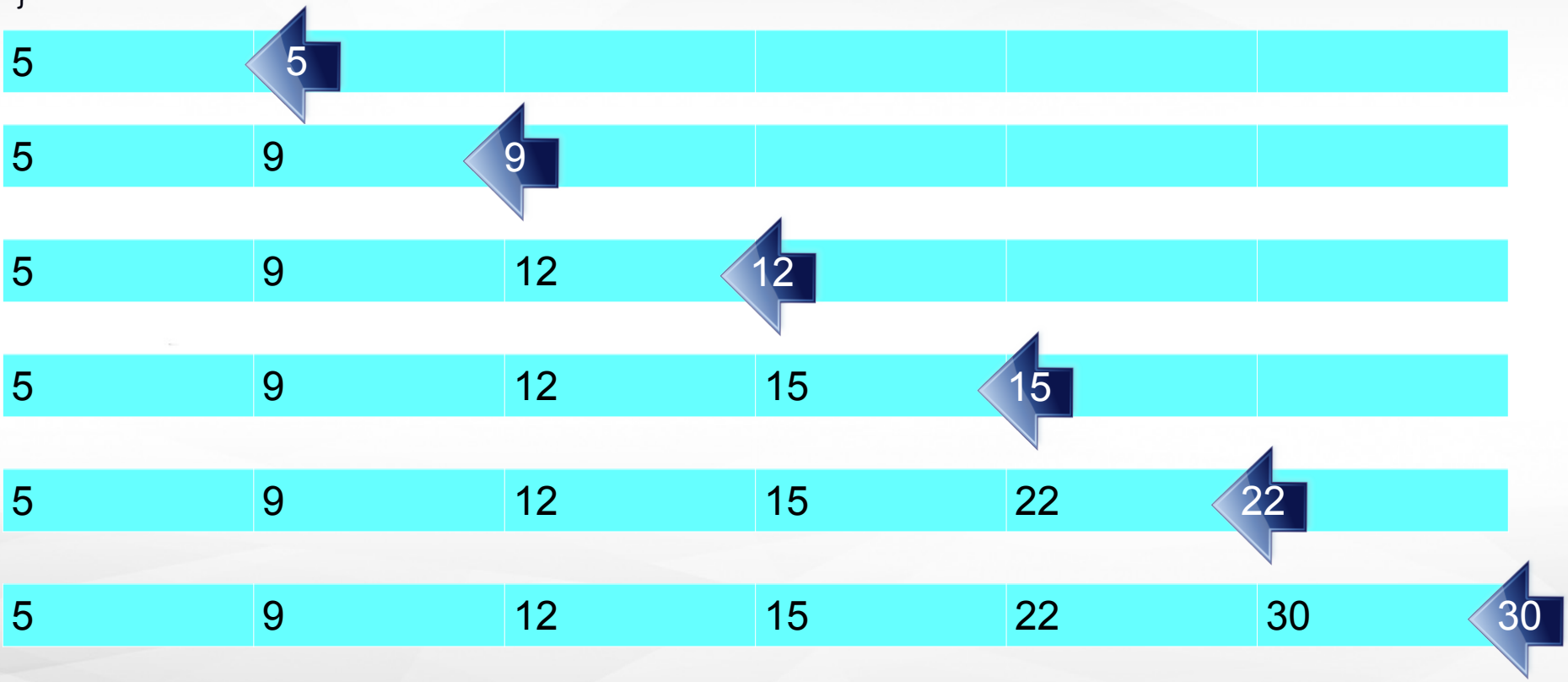
STACK LIFO (LAST IN - FIRST OUT)
push()

```
item = 22;
public void push(int item) {
    items[n++] = item;
}
```



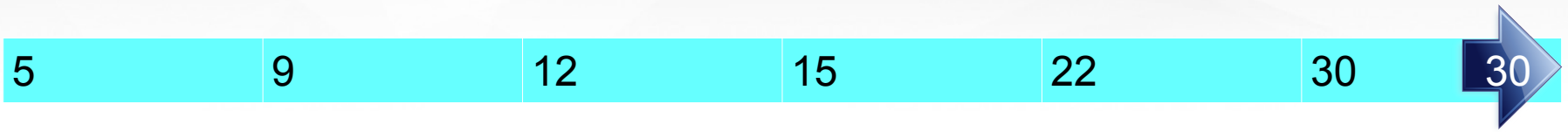
STACK LIFO (LAST IN - FIRST OUT)
push()

```
item = 30;
public void push(int item) {
    items[n++] = item;
}
```



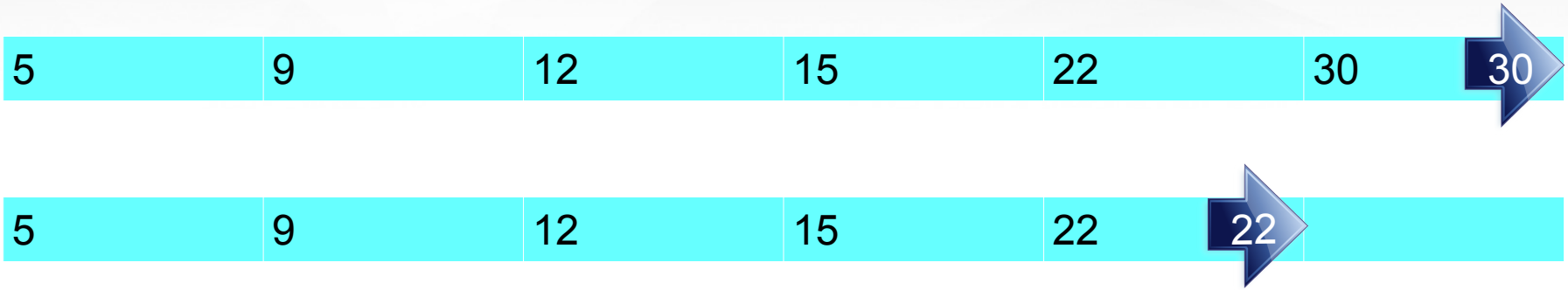
STACK LIFO (LAST IN - FIRST OUT)
push()

```
public String pop() {
        return items[--n];
    }
```



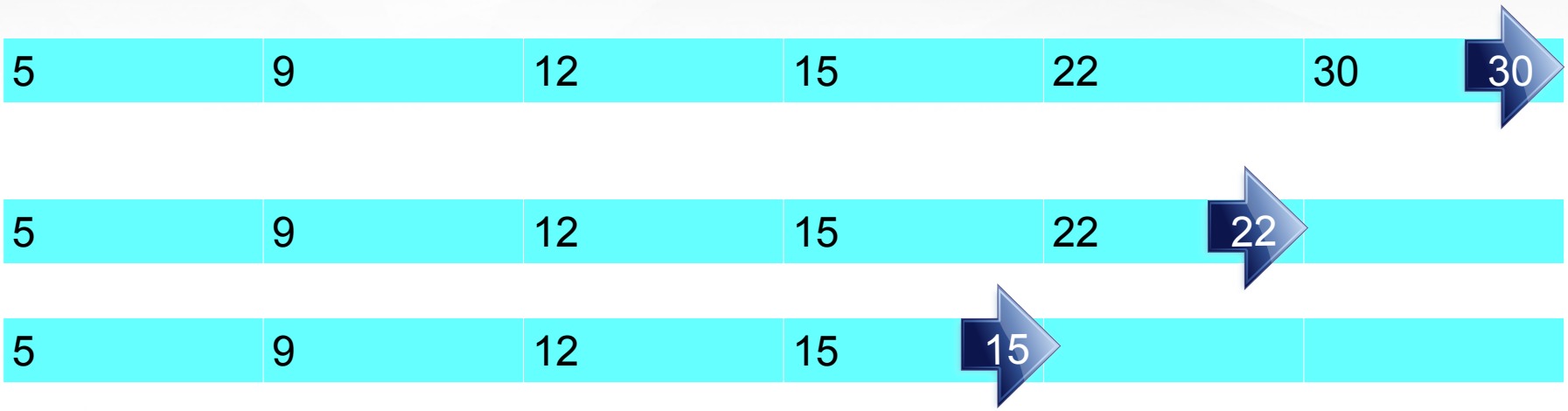
STACK LIFO (LAST IN - FIRST OUT)
push()

```
public String pop() {
        return items[--n];
    }
```



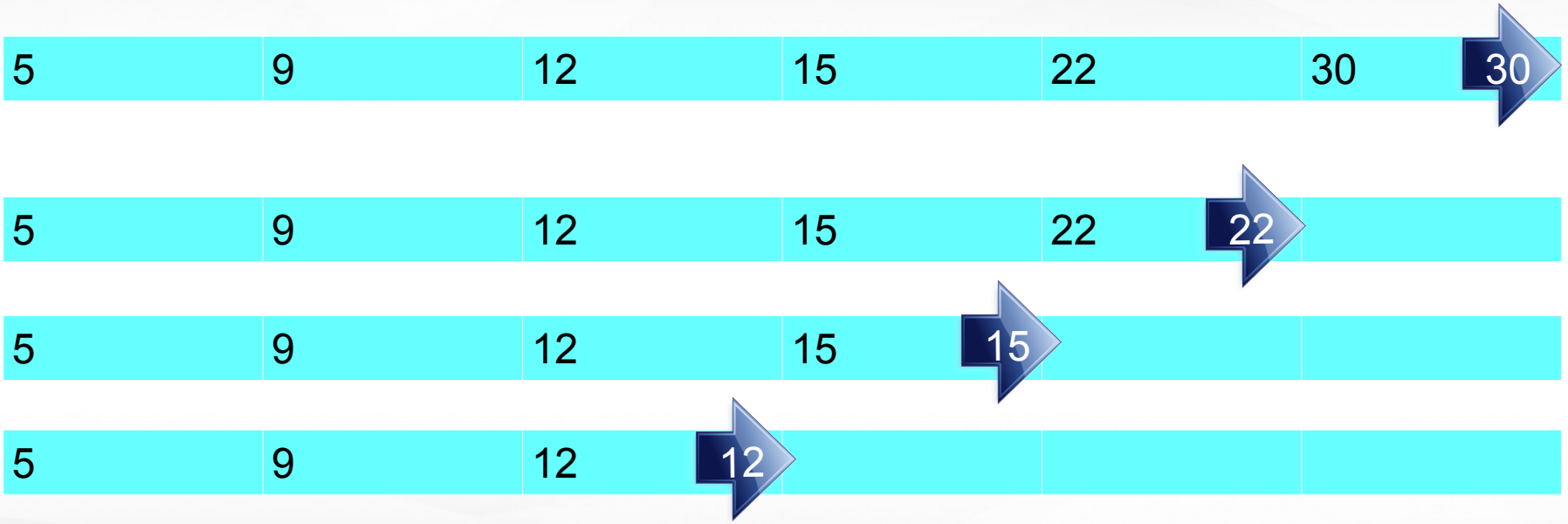
STACK LIFO (LAST IN - FIRST OUT)
push()

```
public String pop() {
        return items[--n];
    }
```



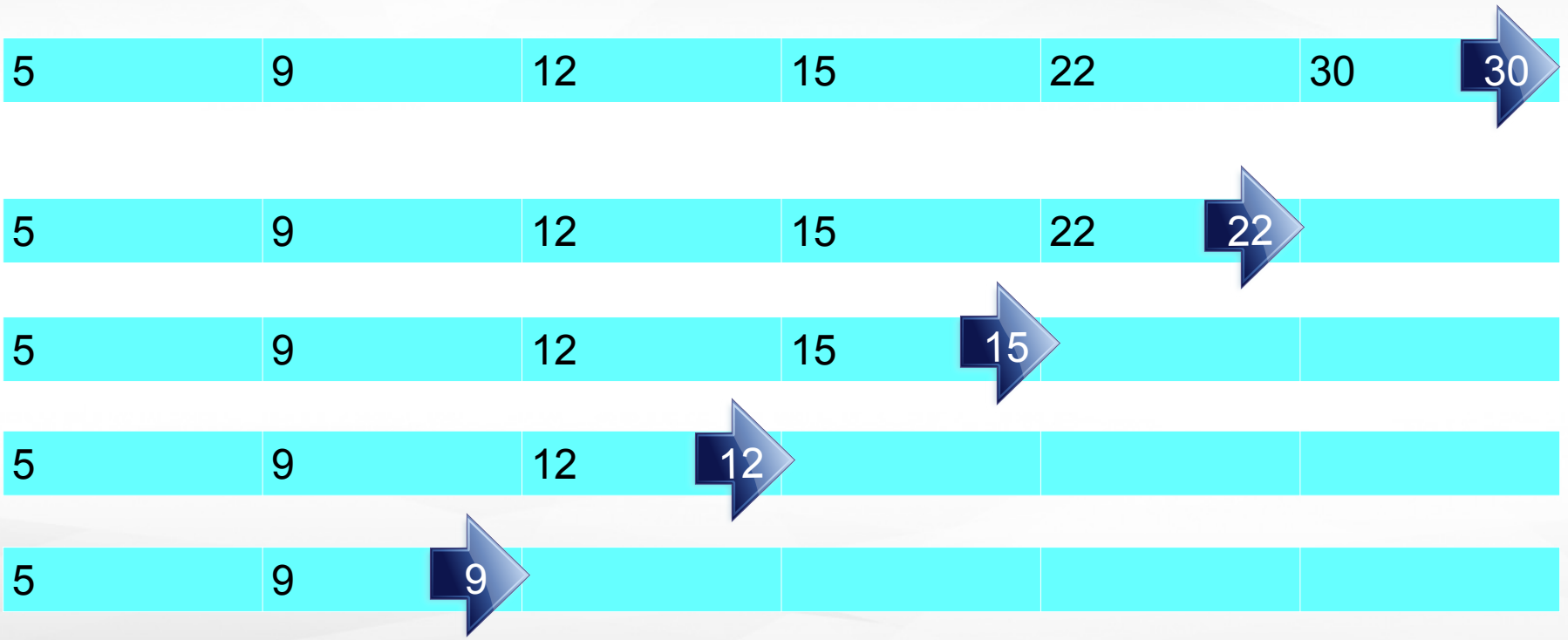
STACK LIFO (LAST IN - FIRST OUT)
push()

```
public String pop() {
        return items[--n];
    }
```



STACK LIFO (LAST IN - FIRST OUT)
push()

```
public String pop() {
        return items[--n];
    }
```



STACK LIFO (LAST IN - FIRST OUT)
push()

```
public String pop() {
        return items[--n];
    }
```