KJSCE CODECELL

STANDARD TEMPLATE LIBRARY WORKSHOP

KJSCE CODECELL

BASICS OF C++

KJSCE CODECELL

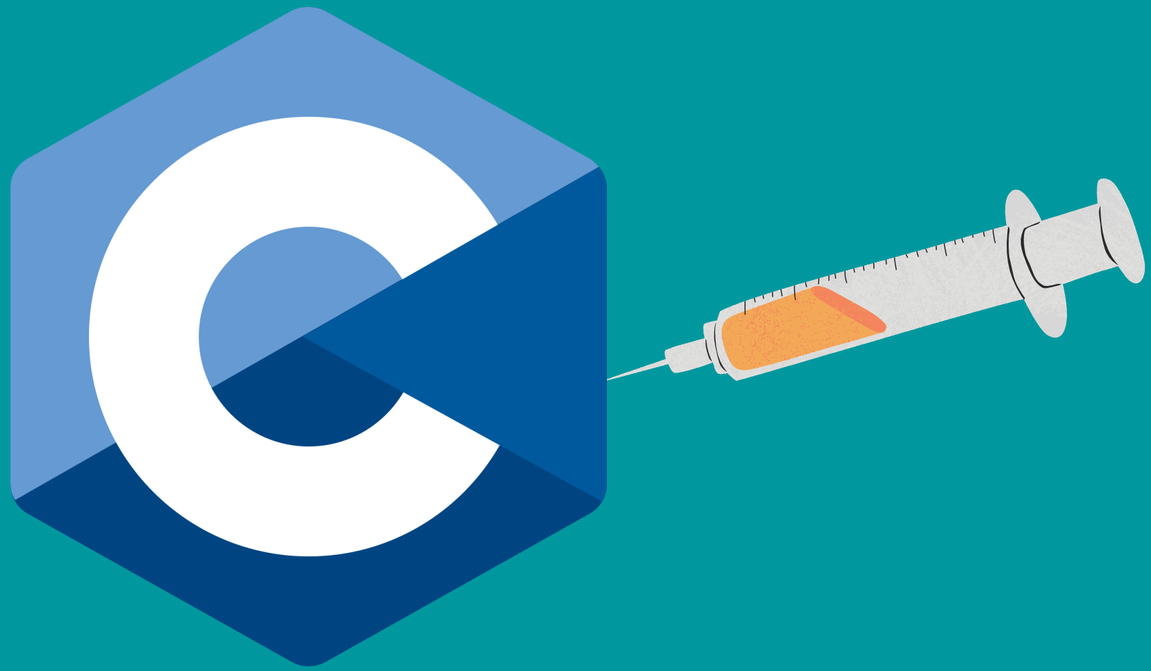
Introduction

C++ is a general purpose programming language which was built as an extension to the C language. So you can not only use the basic functions of C language but also use the benefits of C++ like OOP and STL. Widely used in Competitive Programming and Game Development.

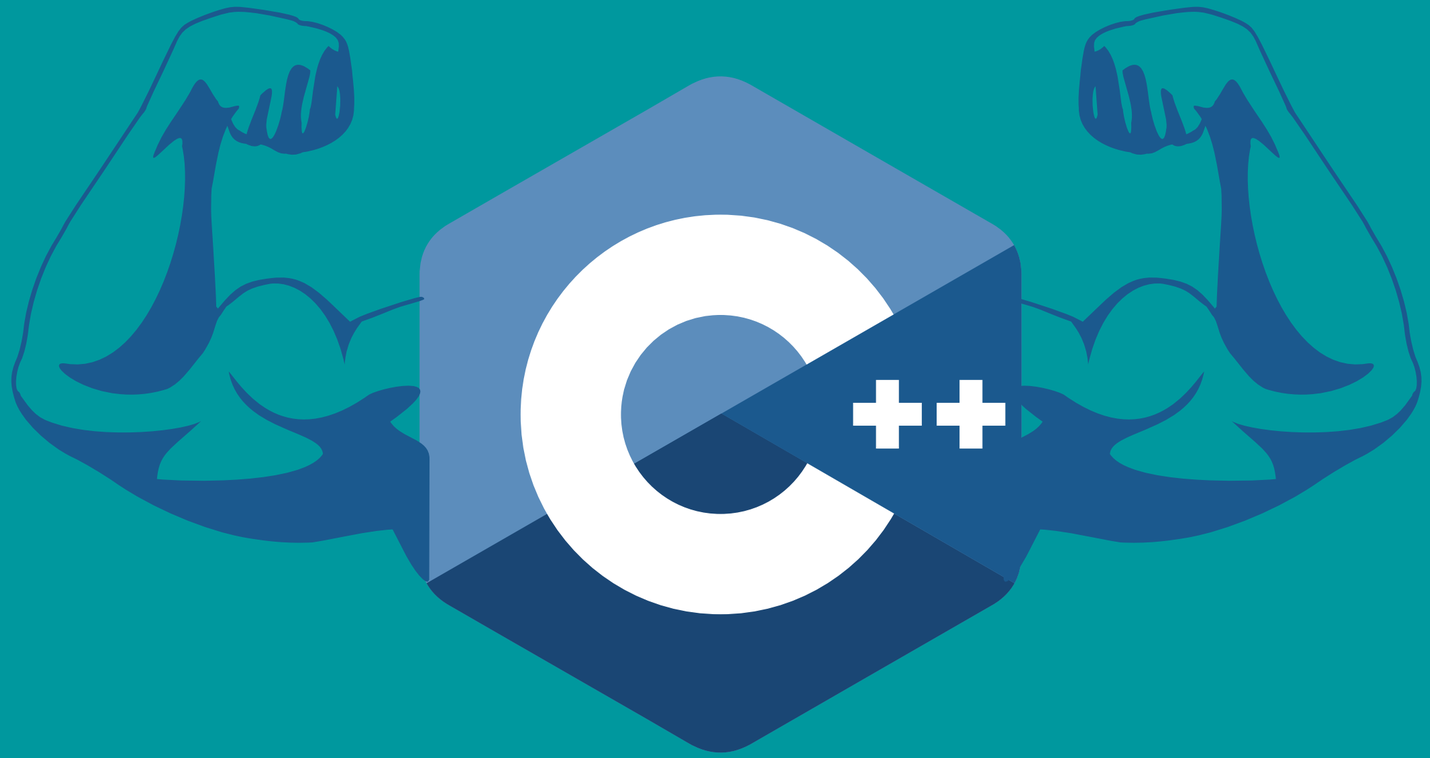
KJSCE CODECELL

Please bhai achhe se samjha de

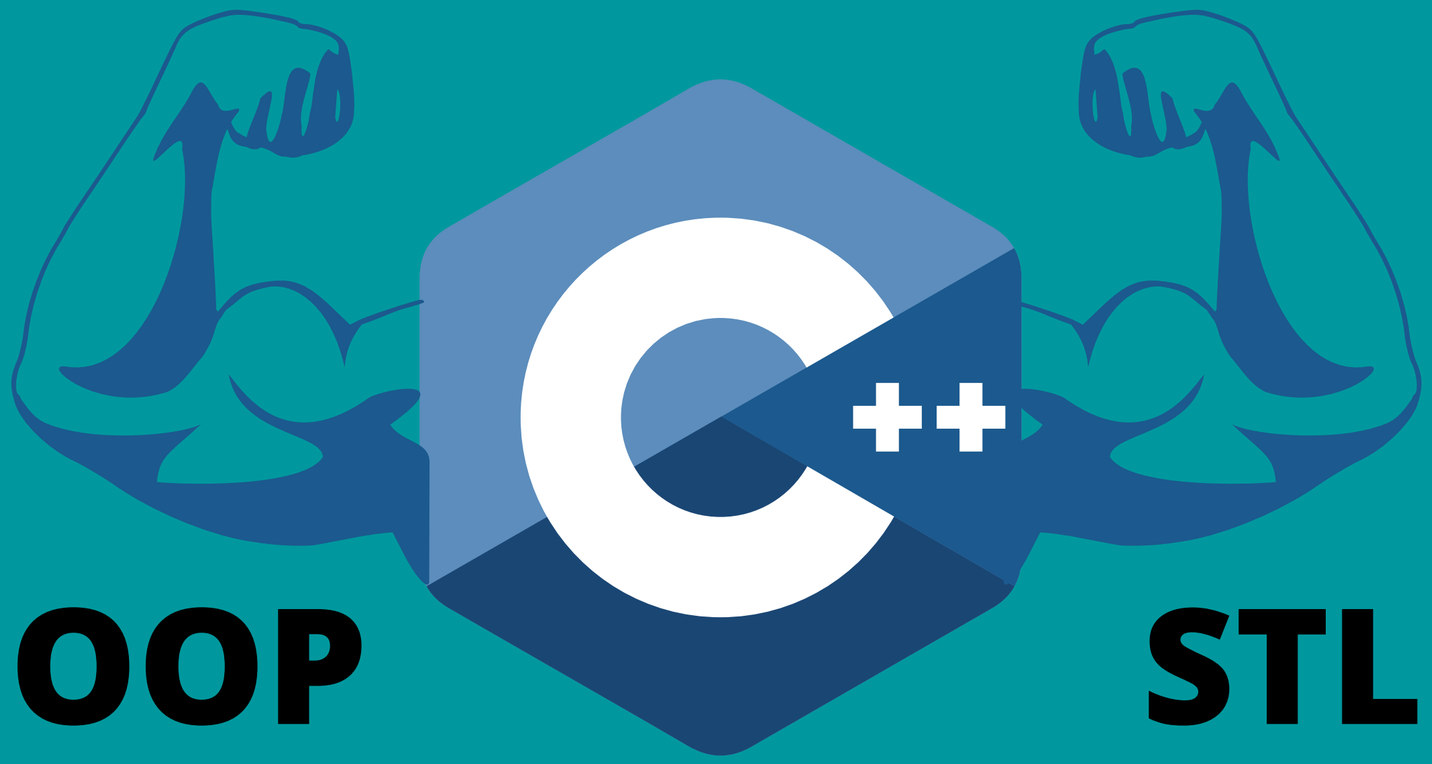
KJSCE CODECELL



KJSCE CODECELL



KJSCE CODECELL



OOP

STL

KJSCE CODECELL

Time to Learn C++

KJSCE CODECELL

Let's Code

https://replit.com/

KJSCE CODECELL

Hello.cpp

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    cout << "Hello World" << endl;
    return 0;
}
```

KJSCE CODECELL

Basic Data Types

- Integers
- Boolean
- Double
- Characters
- String

KJSCE CODECELL

Conditional Statements

If-else are conditional statements, which are generally used when we want to run the code based on some condition.

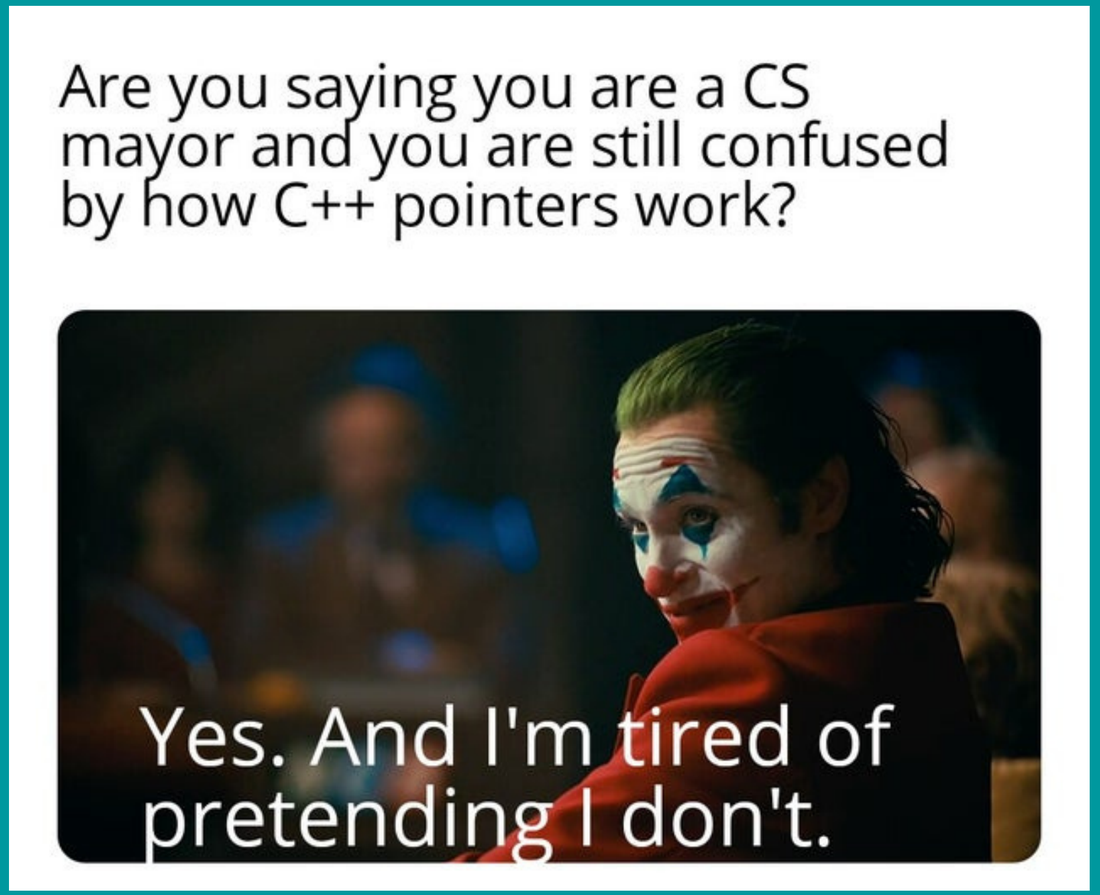
KJSCE CODECELL

Loops

Loop is a sequence of instructions that is continually repeated until a certain condition is reached

KJSCE CODECELL

Pointers



KJSCE CODECELL

Pointers

KJSCE CODECELL

They store in the address of other variables and point towards it.

KJSCE CODECELL

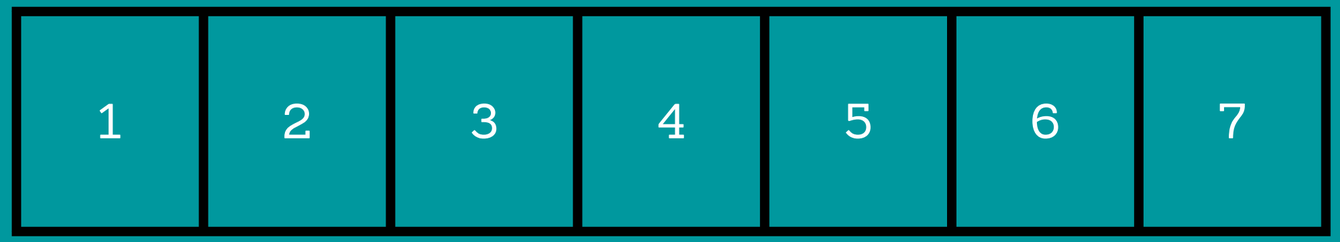
Arrays

Arrays can be defined as group or collection of similar kind of elements under a particular name

They use block of contiguous memory location to store data of **fixed size**

KJSCE CODECELL

arr [7] =



KJSCE CODECELL

Functions

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function.

Functions are used to perform certain actions, and they are important for reusing code.

KJSCE CODECELL

int main()

It serves as the entry point for the program. The computer will start running the code from the beginning of the main function.

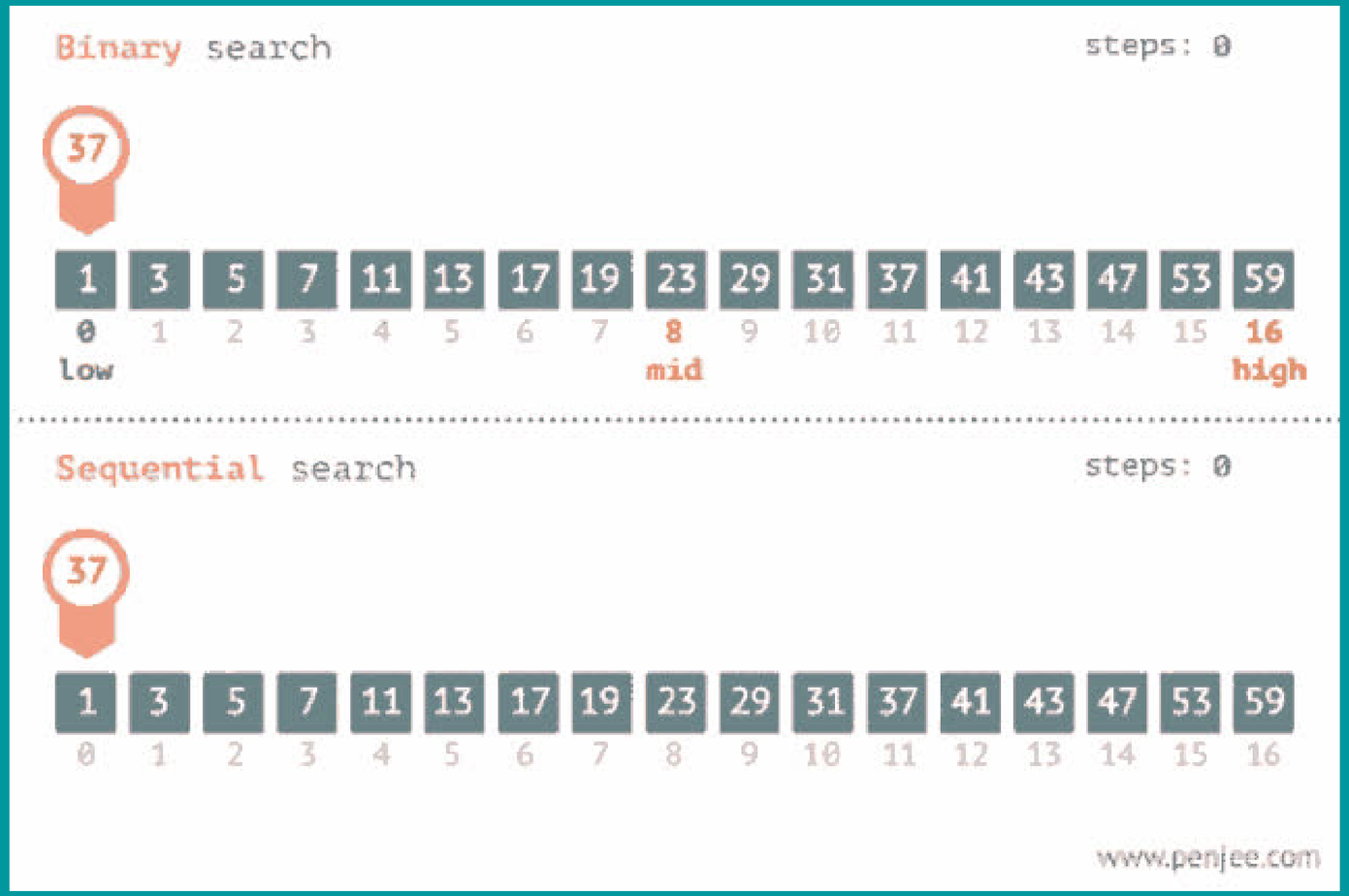
KJSCE CODECELL

binary_search (startAddress , endAddress , valueToFind)

It is a widely used searching algorithm that requires the array to be sorted before search is applied. The main idea behind this algorithm is to keep dividing the array in half (divide and conquer) until the element is found, or all the elements are exhausted.

KJSCE CODECELL

Binary Search Algorithm

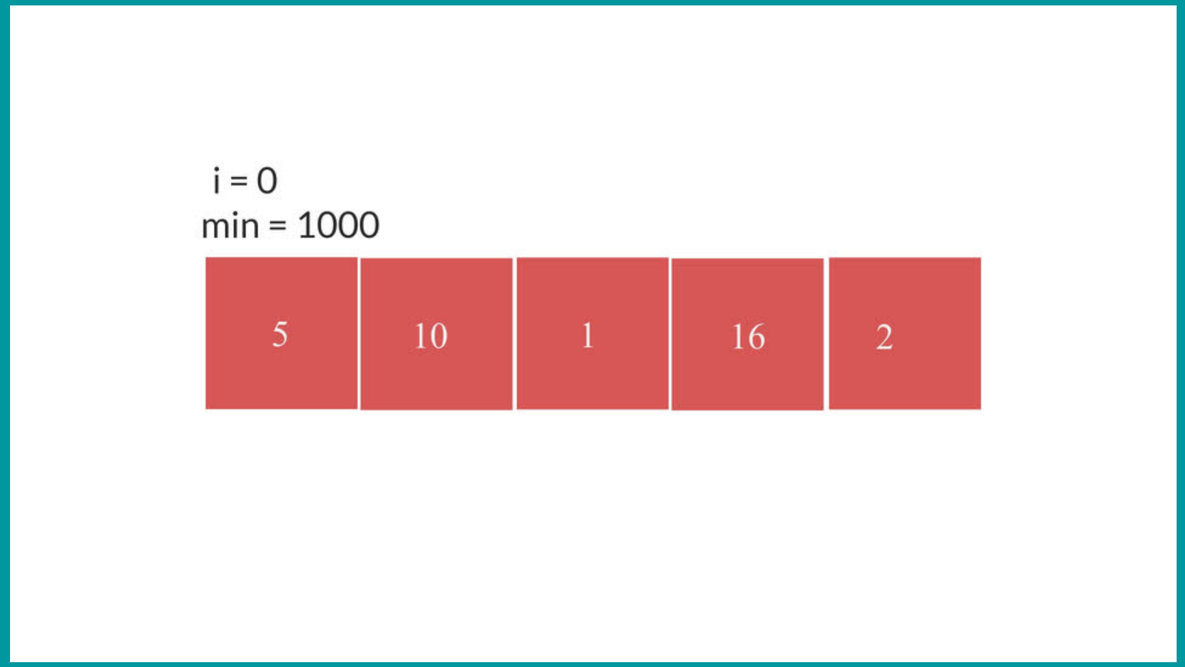


KJSCE CODECELL

Min and Max Element

*min_element (first_index, last_index);

*max_element (first_index, last_index);



accumulate()

```
accumulate(first, last, sum);
```

```
accumulate(first, last, sum, myfun);
```

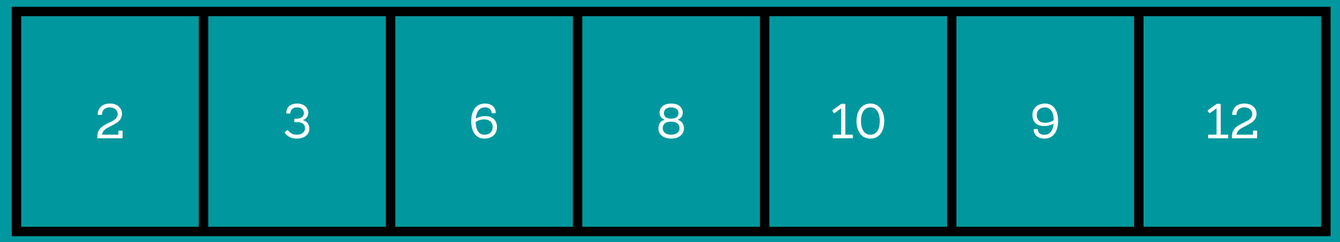
KJSCE CODECELL

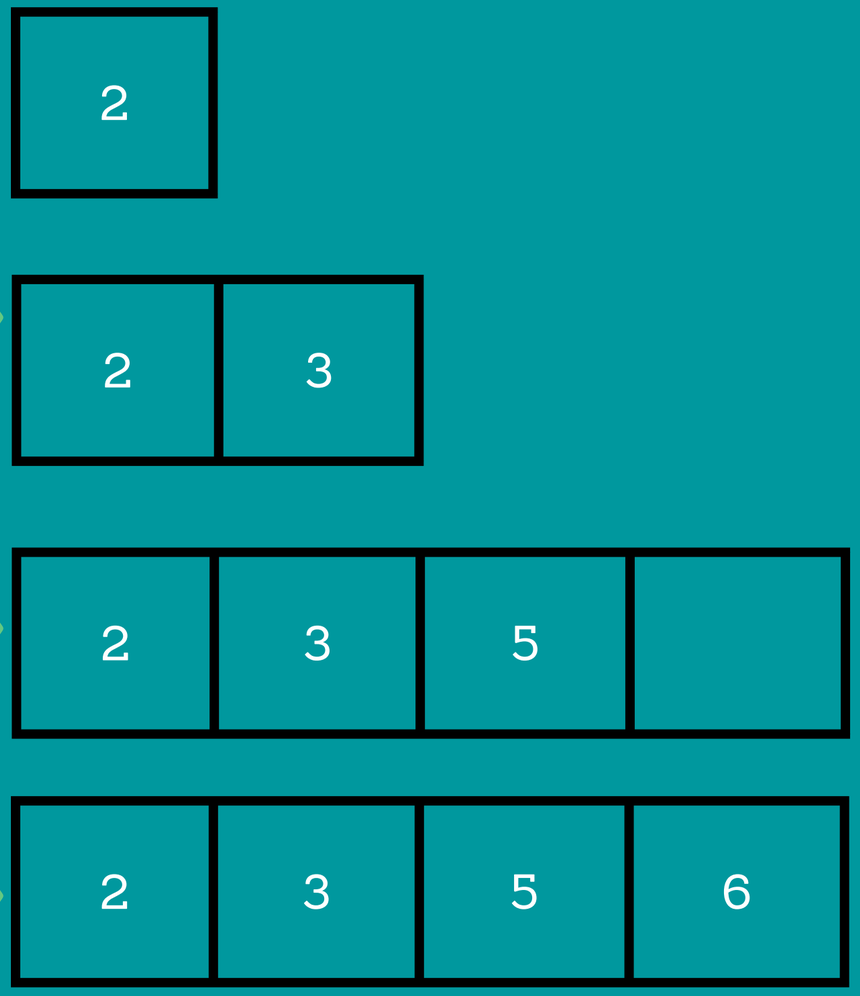
What are vectors?

KJSCE CODECELL

What is the problem with arrays?

arr[7] =





capacity = 1 , size = 1

capacity = 2 , size = 2

capacity = 4 , size = 3

capacity = 4 , size = 4

KJSCE CODECELL

Advantage of doubling capacity

1 -> 1,000

1 -> 2 -> 3 -> 4 -> 5 -> 6
-> 8 ->....1,000

1000 Operations

1 -> 2 -> 4 -> 8 -> 16 ->
32 -> 64 -> 128->
256 -> 512 -> 1024

10 Operations

KJSCE CODECELL

What are pairs?

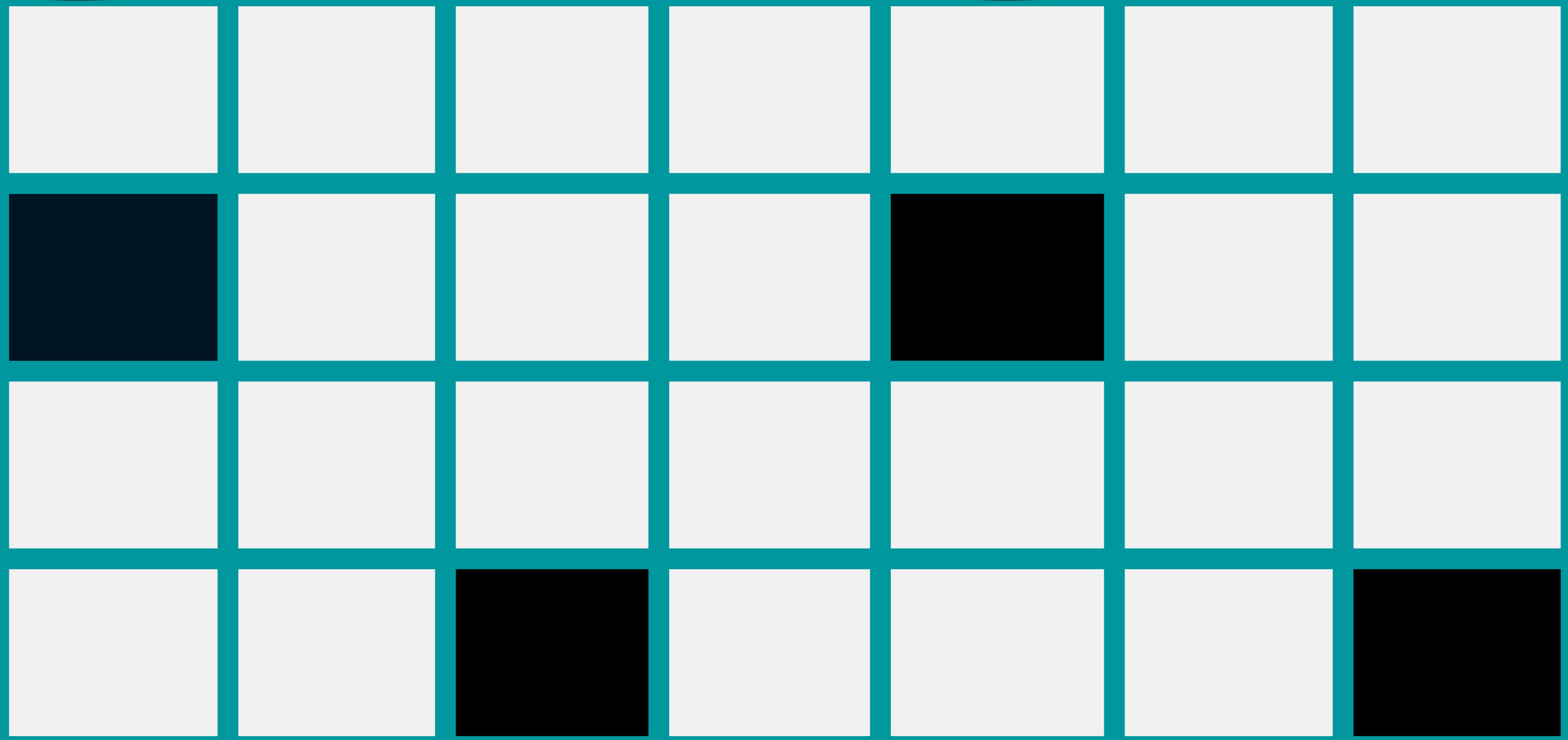
- Pair is used to combine together two values that may be different in type.
- Pair provides a way to store two heterogeneous objects as a single unit.
- The first element is referenced as ‘first’ and the second element as ‘second’ and the order is fixed (first, second).

KJSCE CODECELL

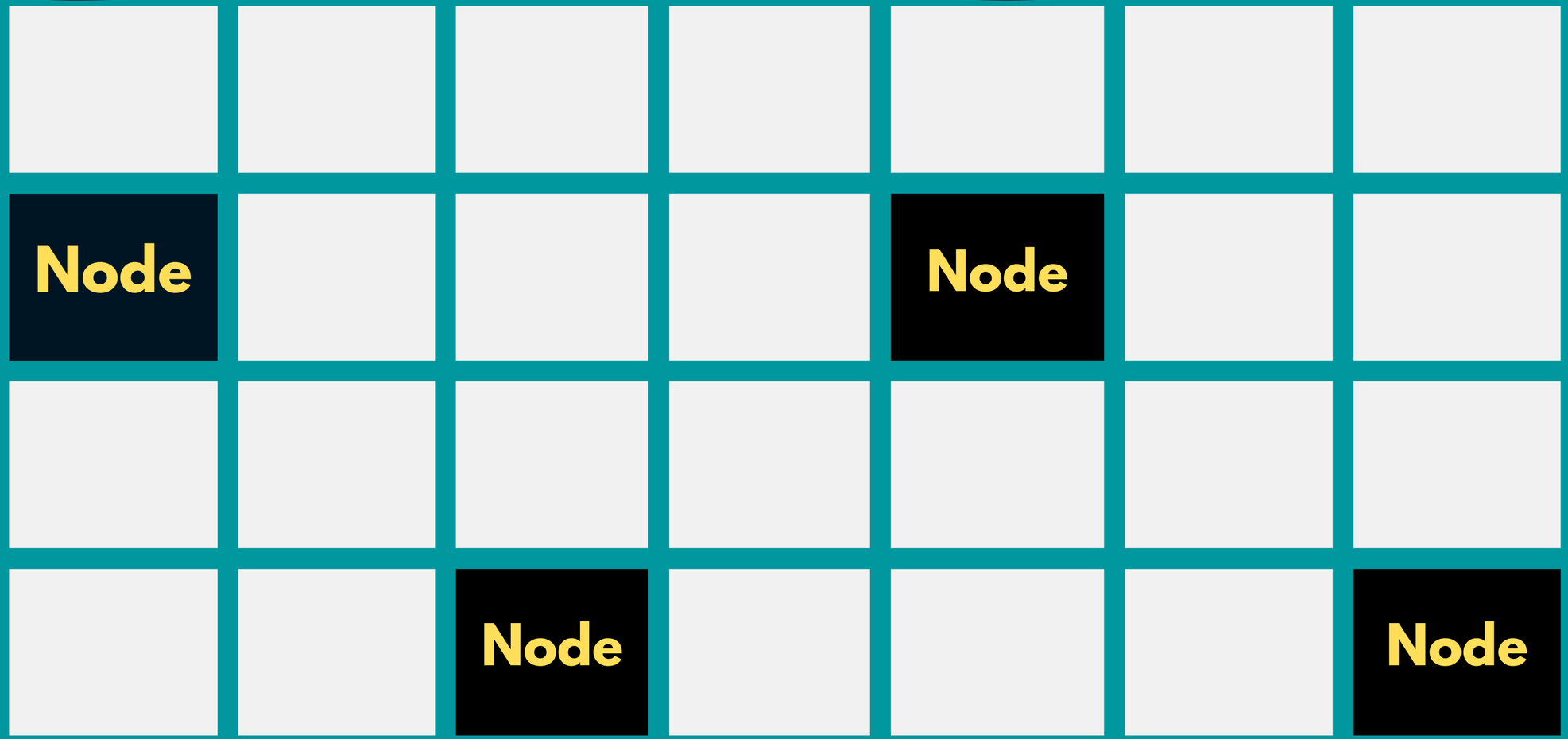
Lists

List is a sequential container class template that allows non-contiguous memory allocation. Due to this, the traversal through the list is slower as compared to vectors. Just like vectors, lists also use iterators to traverse through.

KJSCE CODECELL



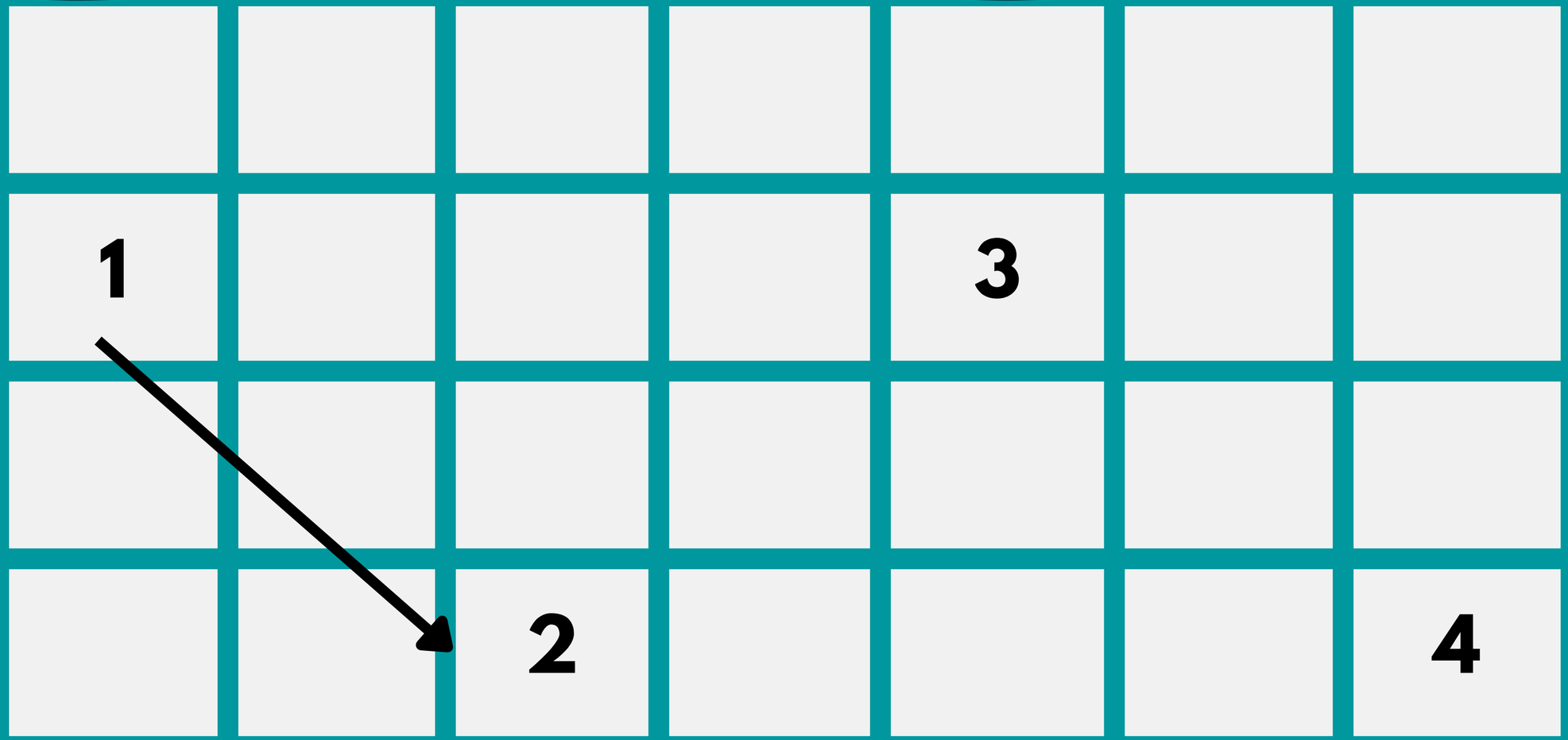
KJSCE CODECELL



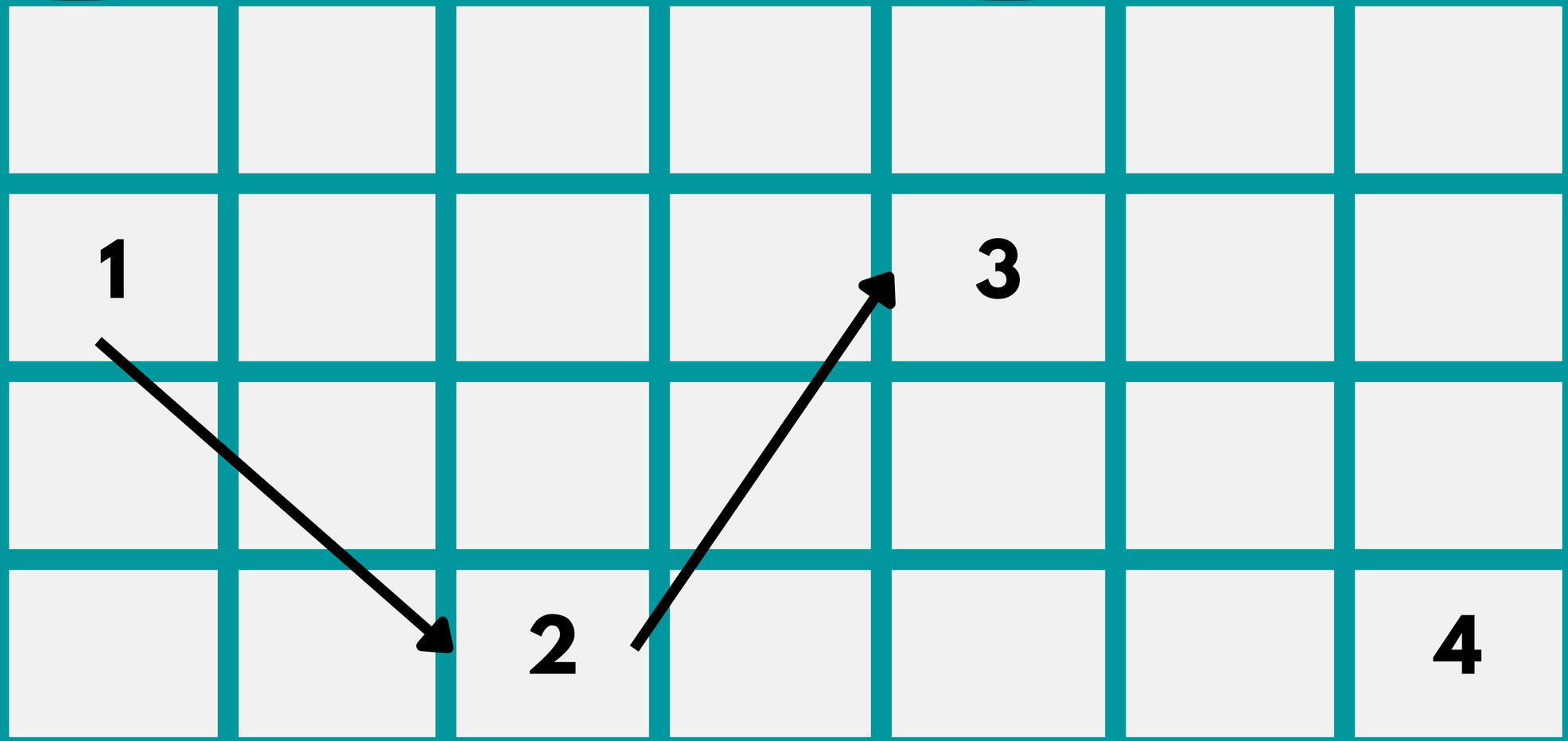
KJSCE CODECELL

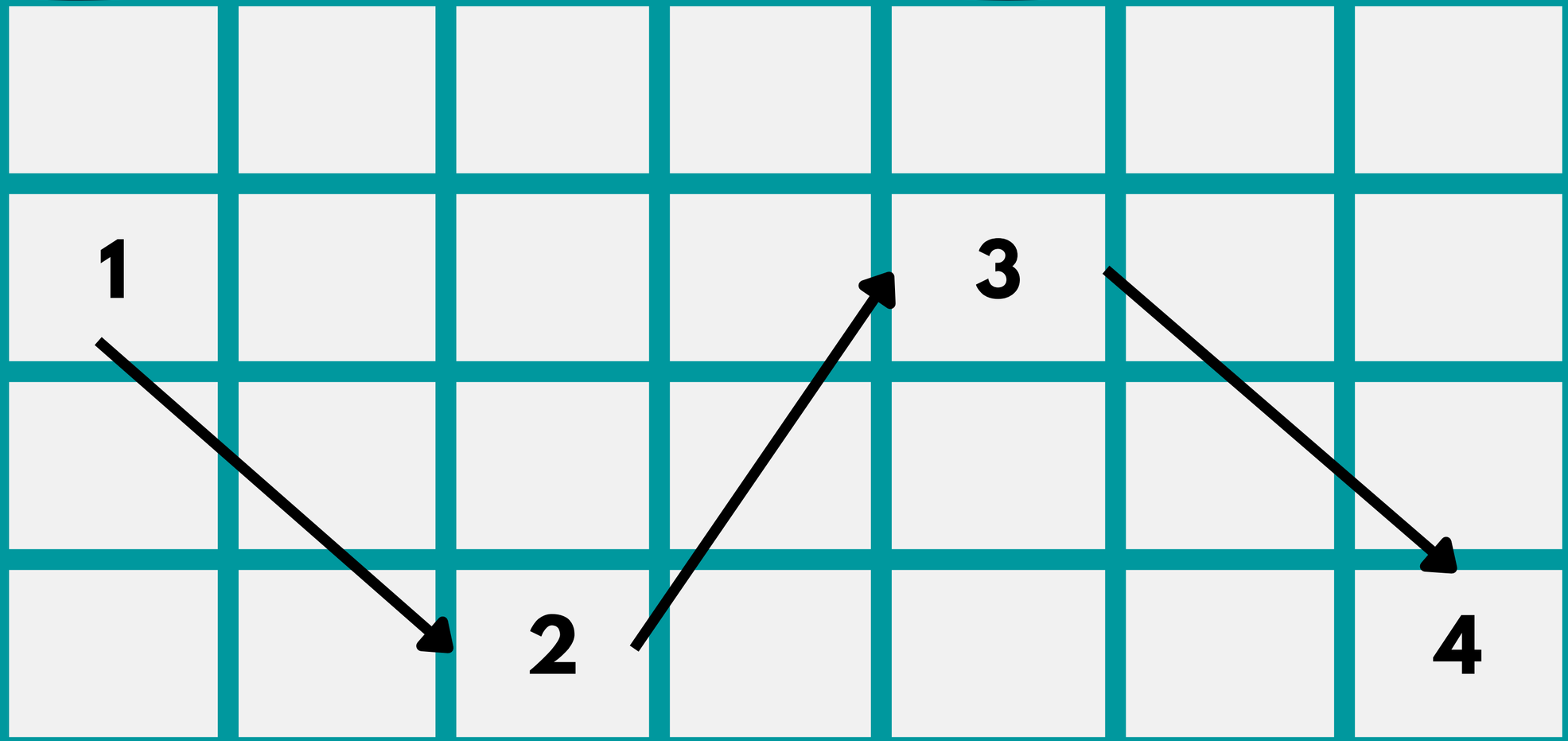
1				3		
		2				4

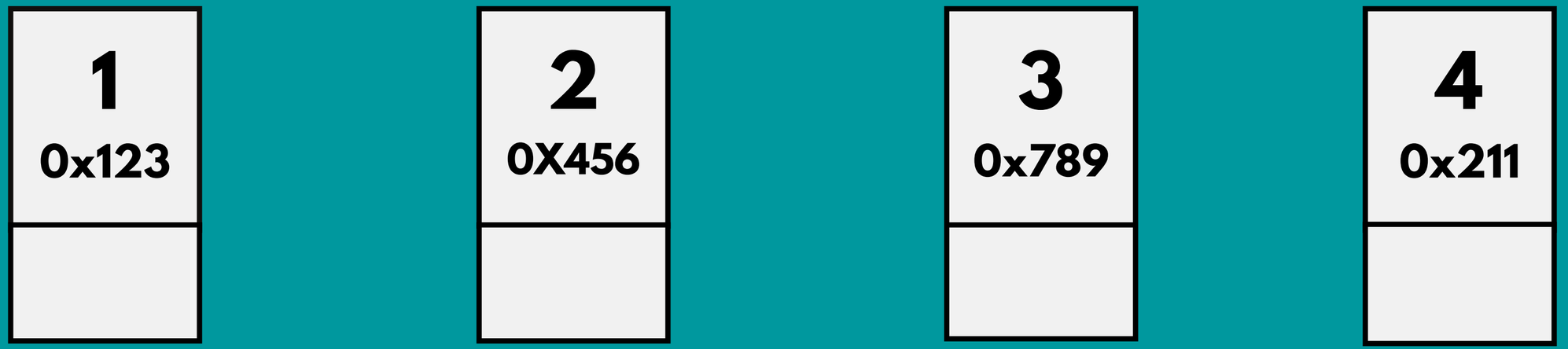
KJSCE CODECELL

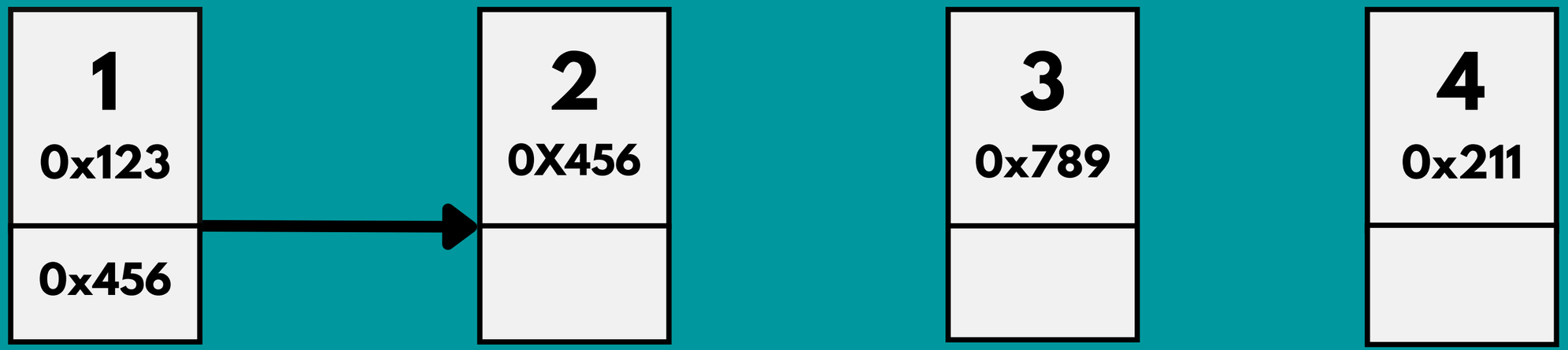


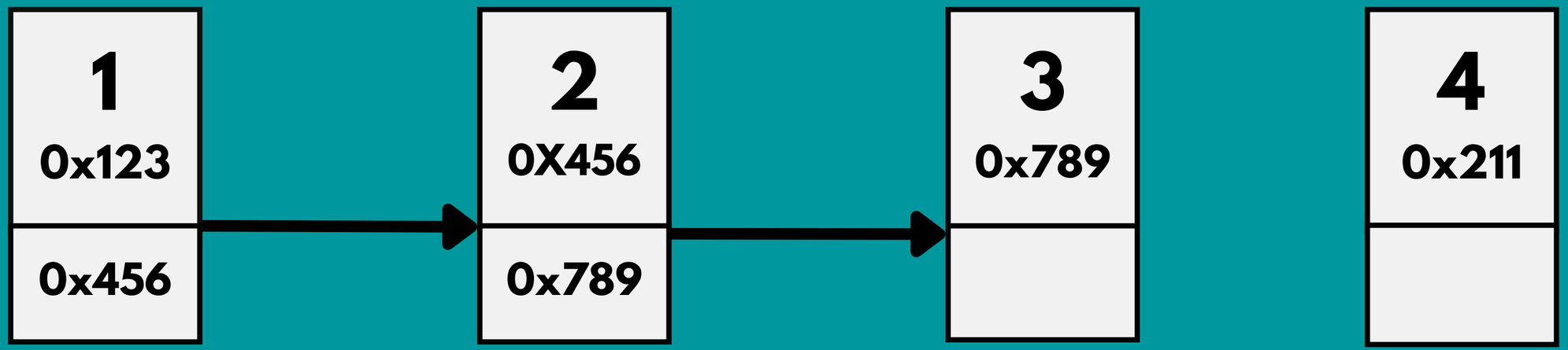
KJSCE CODECELL

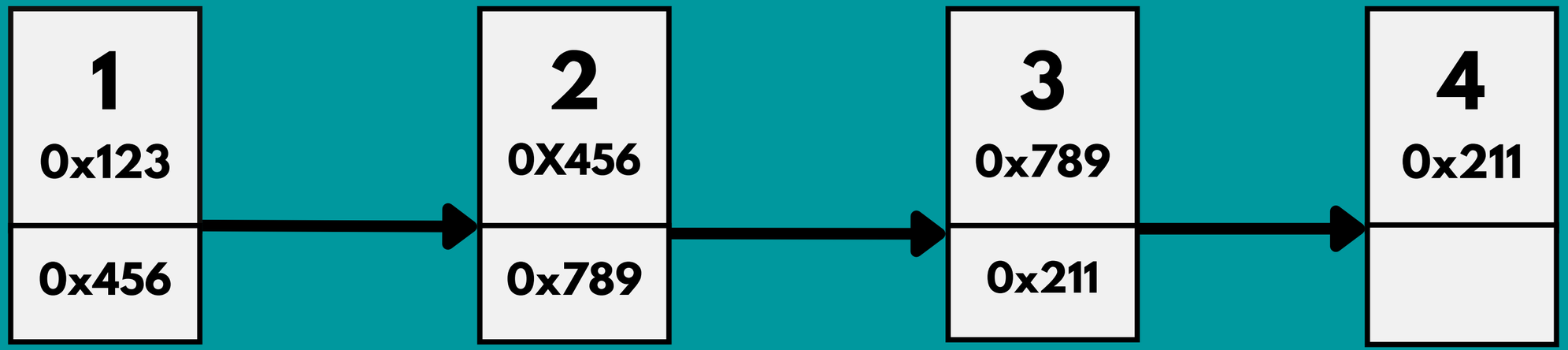



1
2
3
4


1
0x123
2
0X456
3
0x789
4
0x211

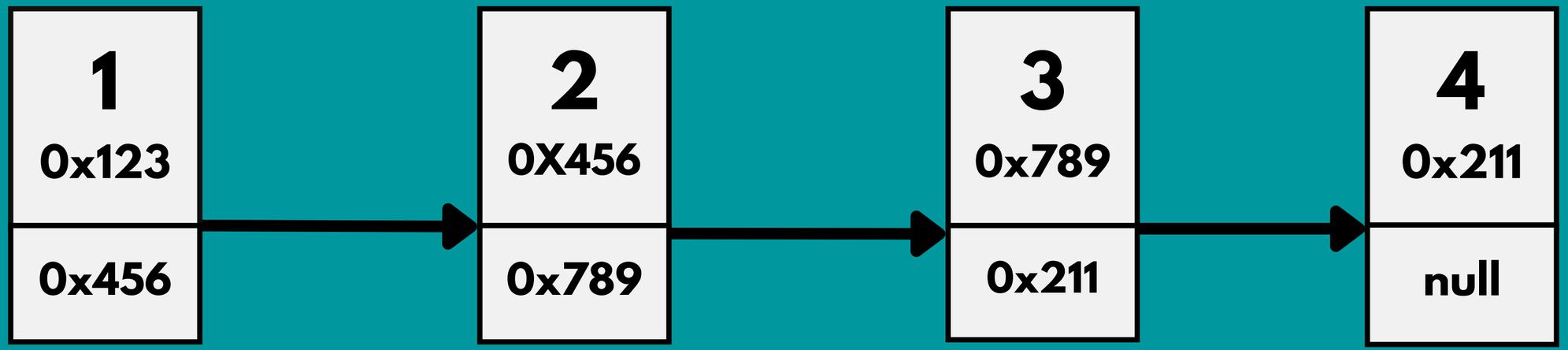

1
0x123
0x456
2
0X456
3
0x789
4
0x211


1
0x123
0x456
2
0X456
0x789
3
0x789
4
0x211


1
0x123
0x456
2
0X456
0x789
3
0x789
0x211
4
0x211

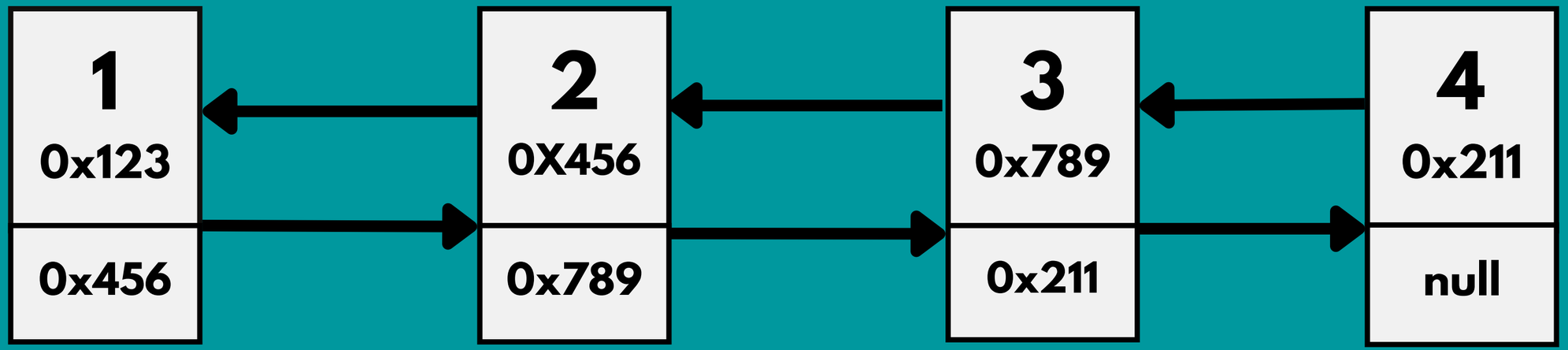
KJSCE CODECELL

Singly Linked List



KJSCE CODECELL

Doubly Linked List



KJSCE CODECELL

Lists

Arrays are better for random access

Linked Lists are better for quick insertion and deletion

KJSCE CODECELL

SETS

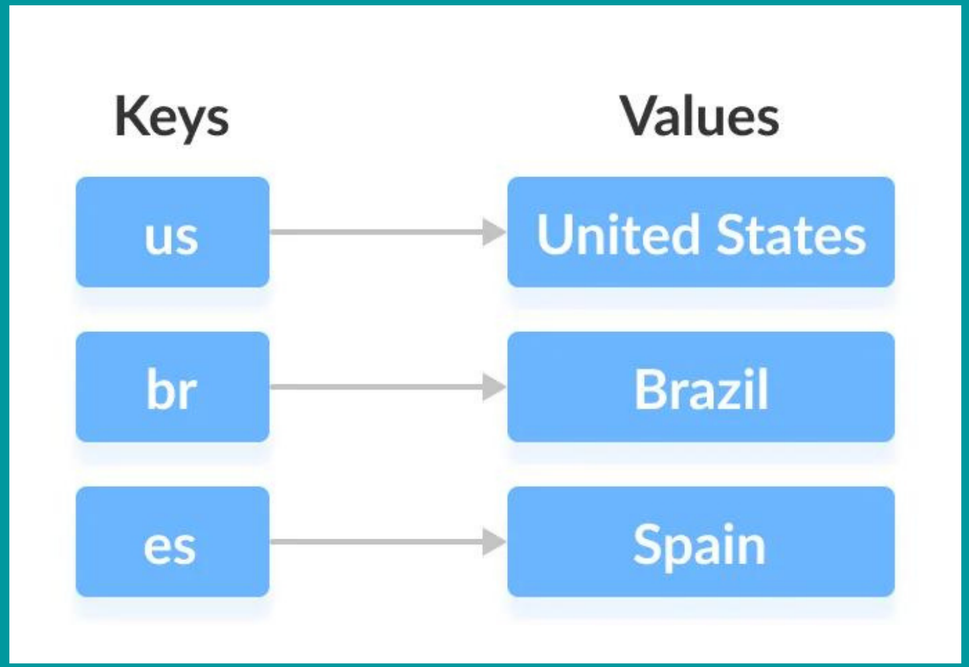
Properties:

- Each element is unique.
- Can contain any data type.
- Values are stored in sorted order.
- The values are immutable i.e the values of the set cannot be modified.
- The values in a set are unindexed.

KJSCE CODECELL

Maps

Maps are associative containers that store elements formed by a combination of a key value and a mapped value.



KJSCE CODECELL

Why Maps?

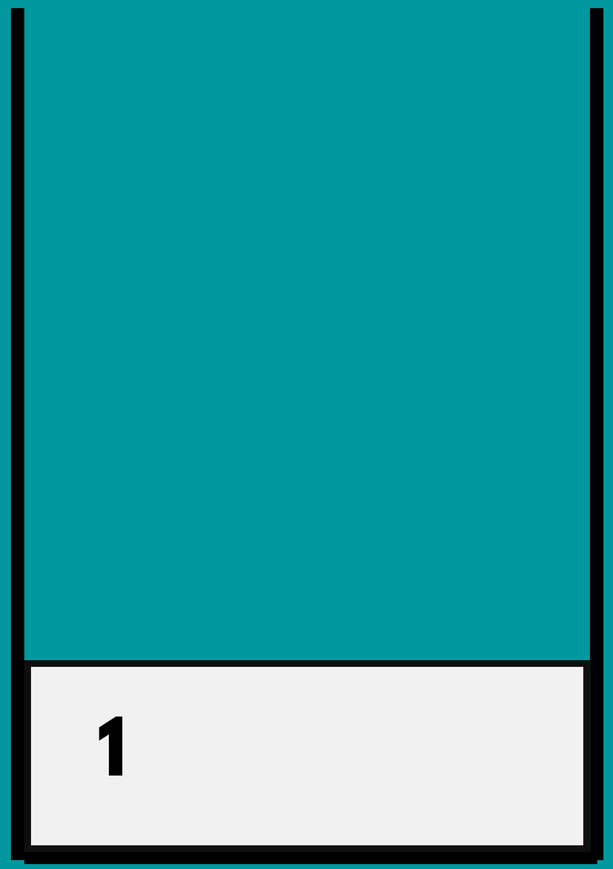
- They store unique key values and therefore, the keys are in a sorted order.
- Only one value for every key.
- Map can be used as an associative array.

KJSCE CODECELL

Stack

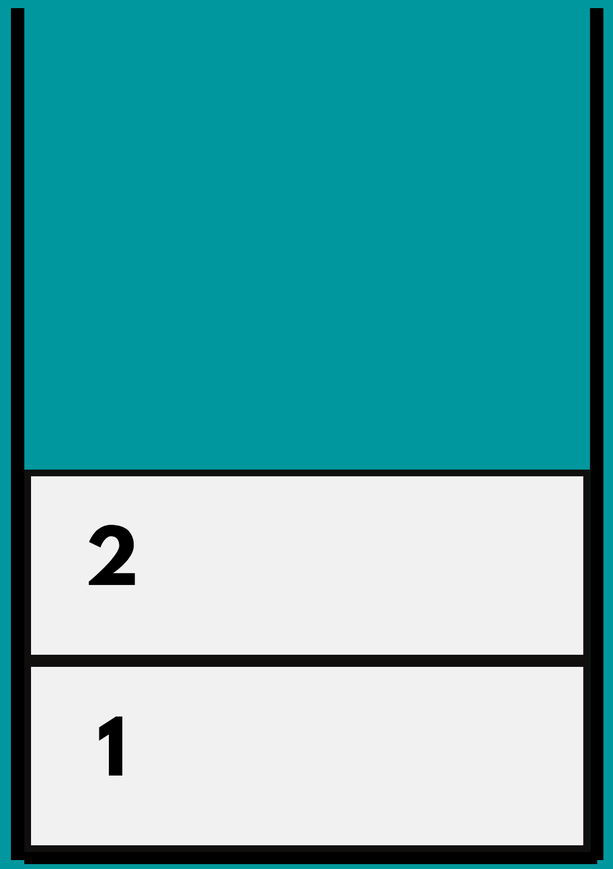
Stacks belong to a type of container adaptors which follow LIFO(Last In First Out) principle[in lifo, the element is added and removed from the same end]. Container adaptors are classes that rely on an object of one of the other container classes(such as list, vector) to handle the elements.

KJSCE CODECELL



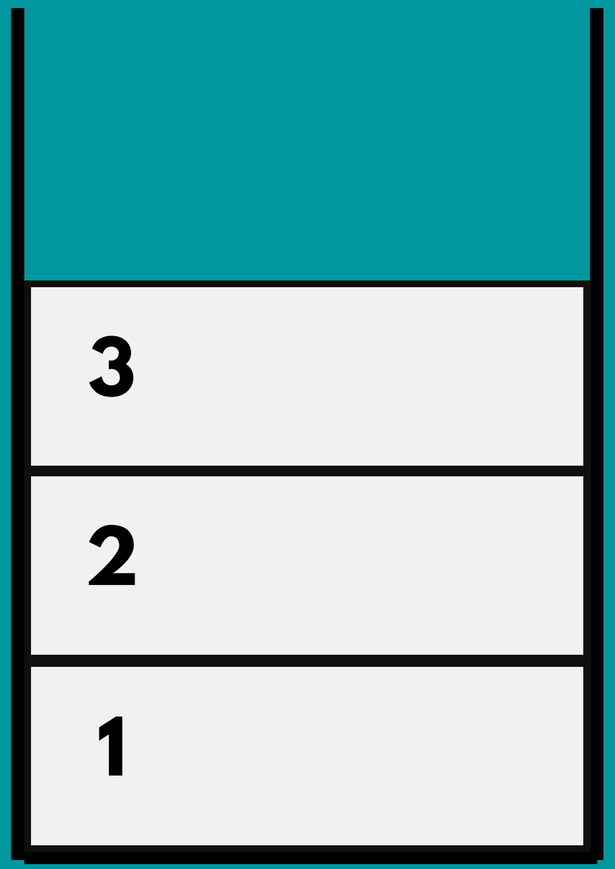
Stack.push(1)

KJSCE CODECELL



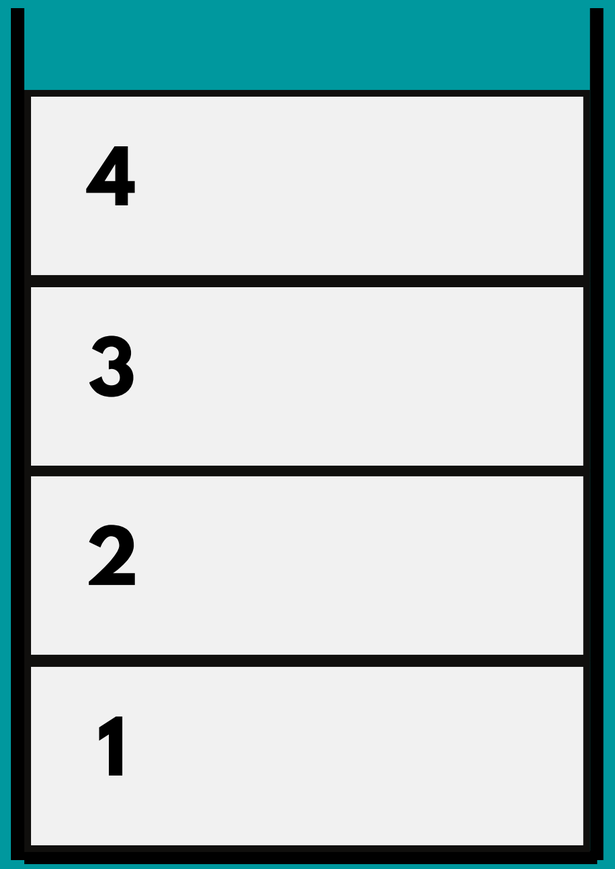
Stack.push(2)

KJSCE CODECELL

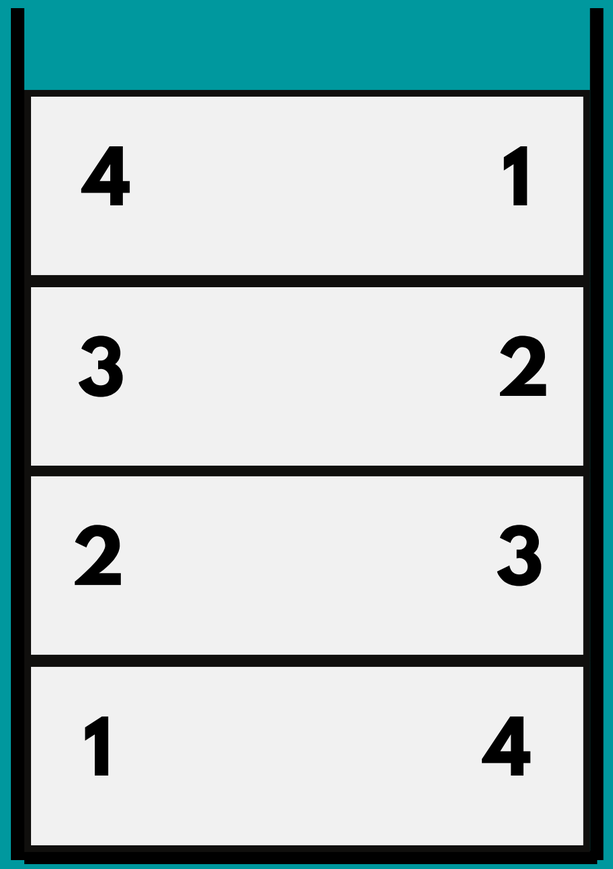


Stack.push(3)

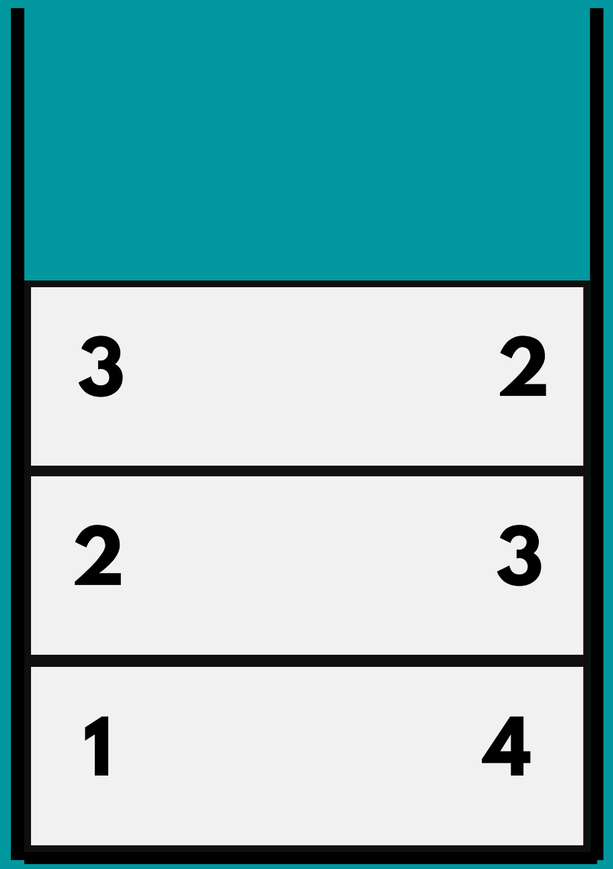
KJSCE CODECELL



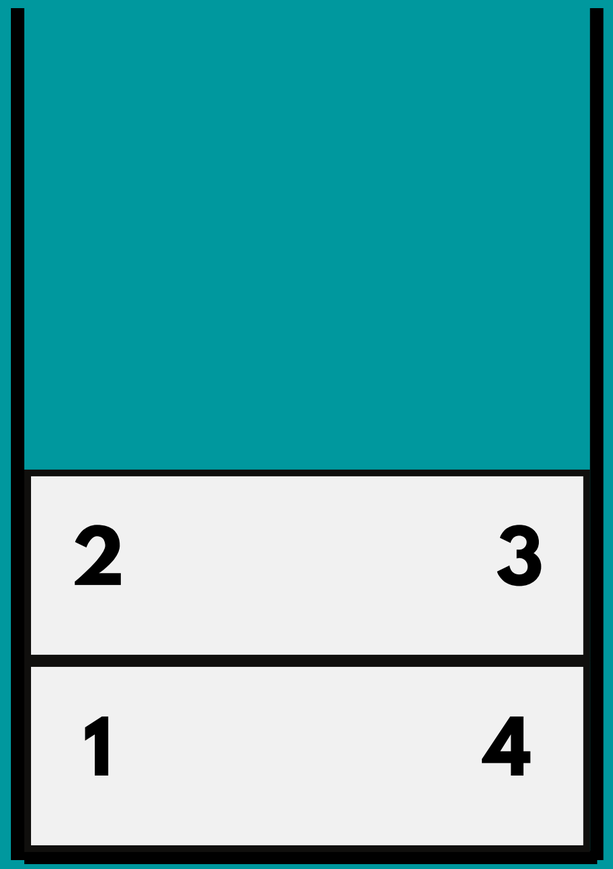
Stack.push(4)



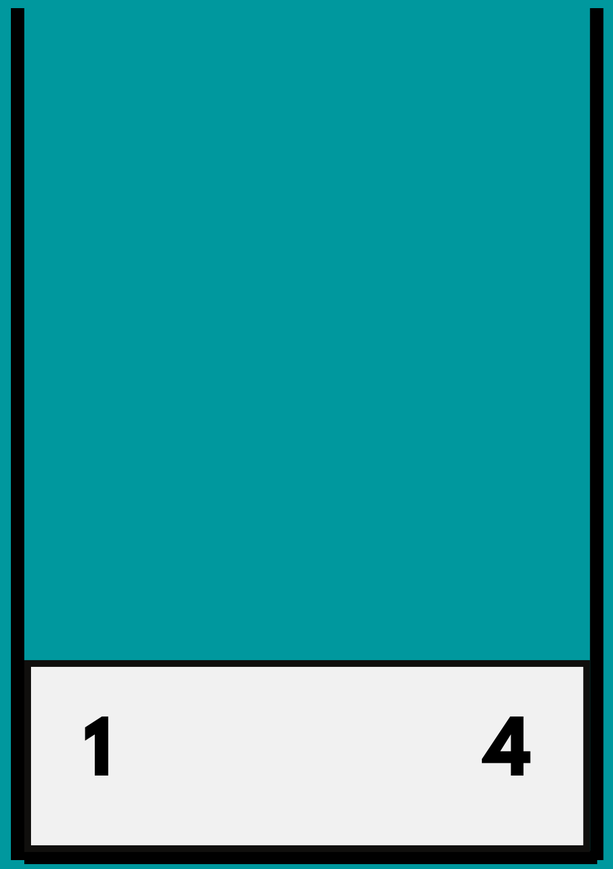
LIFO - Last in first out



LIFO - Last in first out

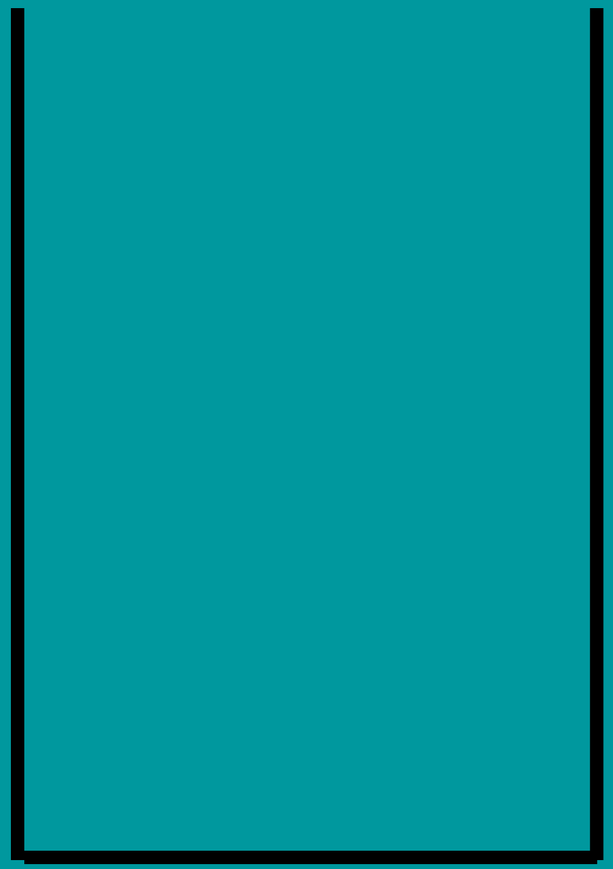


LIFO - Last in first out



LIFO - Last in first out

KJSCE CODECELL



LIFO - Last in first out

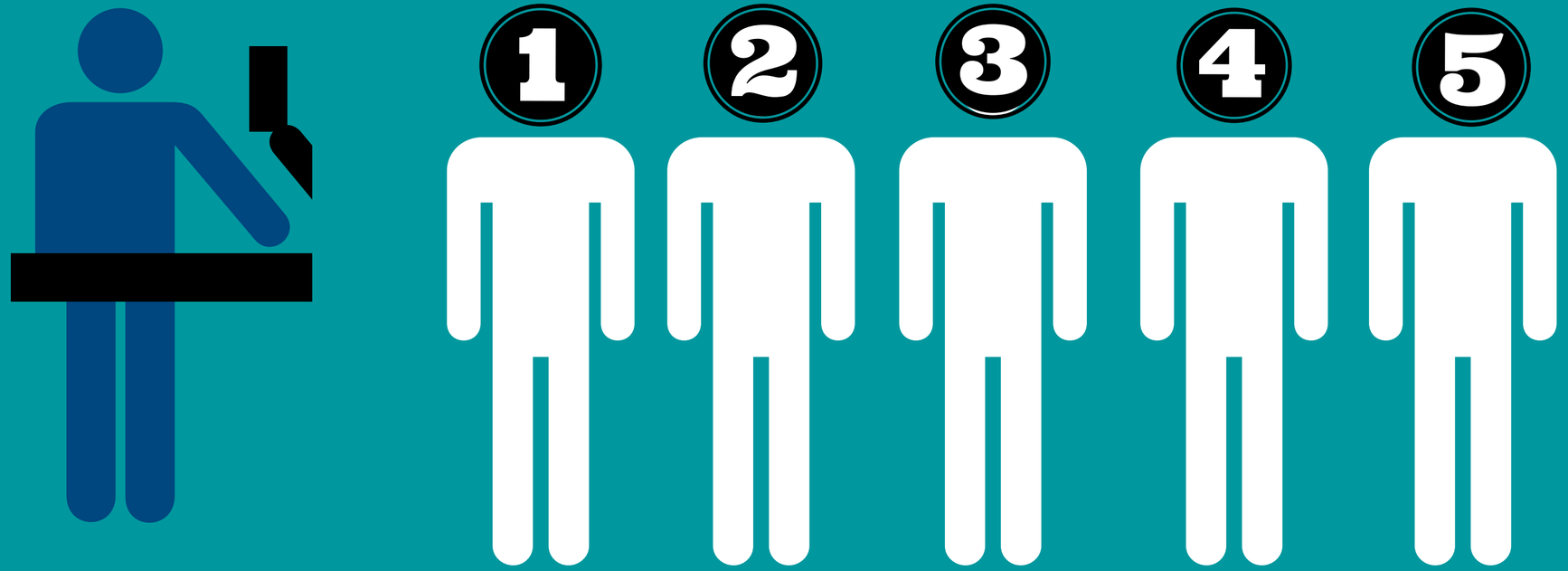
KJSCE CODECELL

Queue

Queues are a type of container adapter, specifically designed to operate on FIFO context i.e. first in first out context , where elements are inserted into one end of the container and extracted from the other.

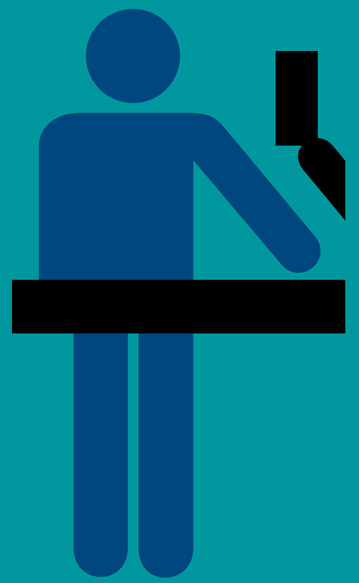
KJSCE CODECELL

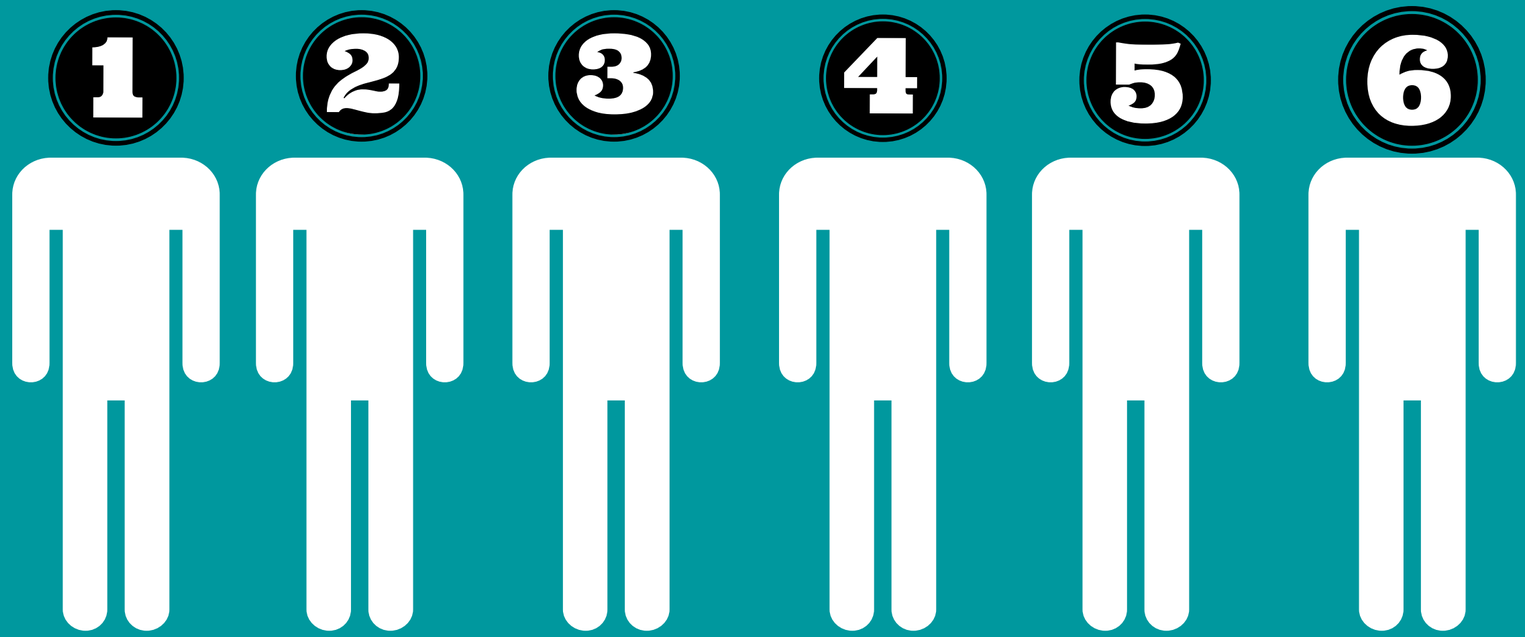
Queue



KJSCE CODECELL

FIFO - First in first out





KJSCE CODECELL

FIFO - First in first out