

# Exploring Swift Collections

NSLondon 2022.1

# Why create “Swift Collections”?

**“The goal of the Swift project is to create the best available language for uses ranging from systems programming, to mobile and desktop apps, scaling up to cloud services.”**

Apple

# What is the “Swift Collections” Package?

**“A new open-source package focused on extending the set of available Swift data structures.”**

Apple

# Why should I care?

- Great performance
- Safe (reliable) code
- Easy to write
- Easy to read
- No third-party dependencies
- Swifty code and APIs



# Data Types Overview

- Deque
- OrderedSet
- OrderedDictionary

# What is a Collection?

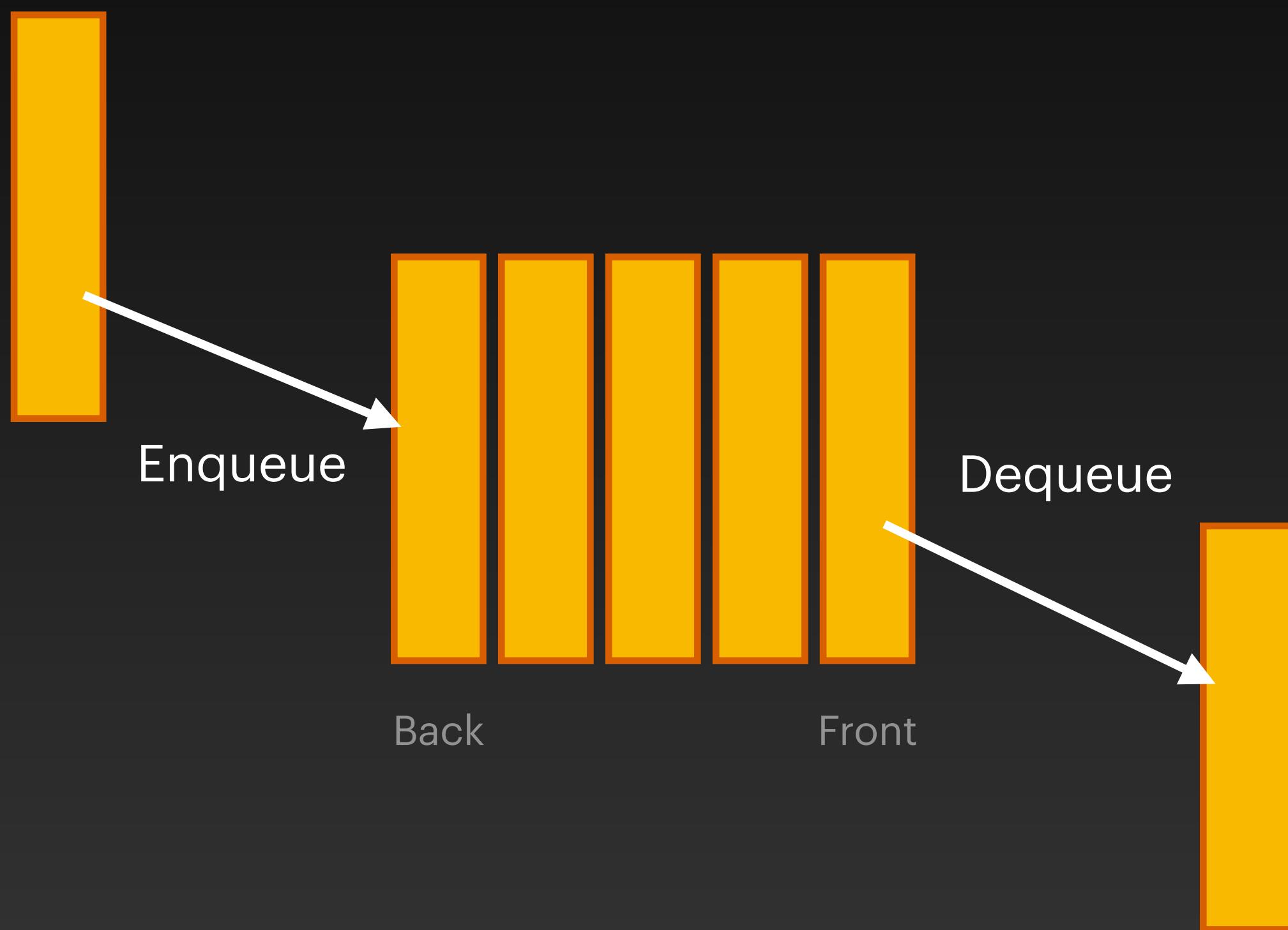
**“A sequence whose elements can be traversed multiple times, nondestructively, and accessed by an indexed subscript.”**

Apple

**This is just the beginning!**

# Queues

- Queues are First In First Out (FIFO).



# DispatchQueue

Grand Central Dispatch uses Queues too!

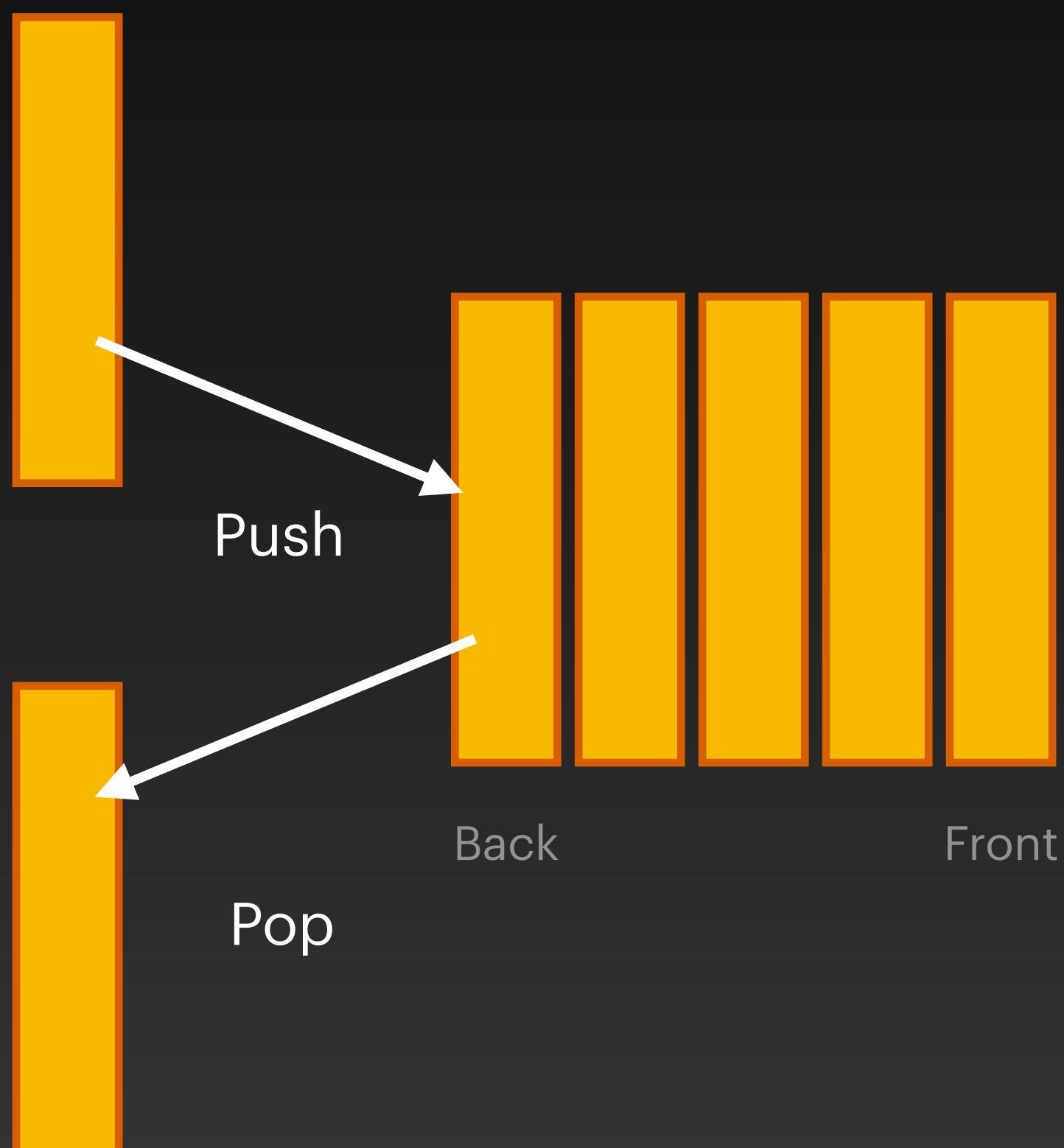
```
DispatchQueue.main.async { [weak self] in  
    print(self?.colors)  
}
```



image: Flaticon.com

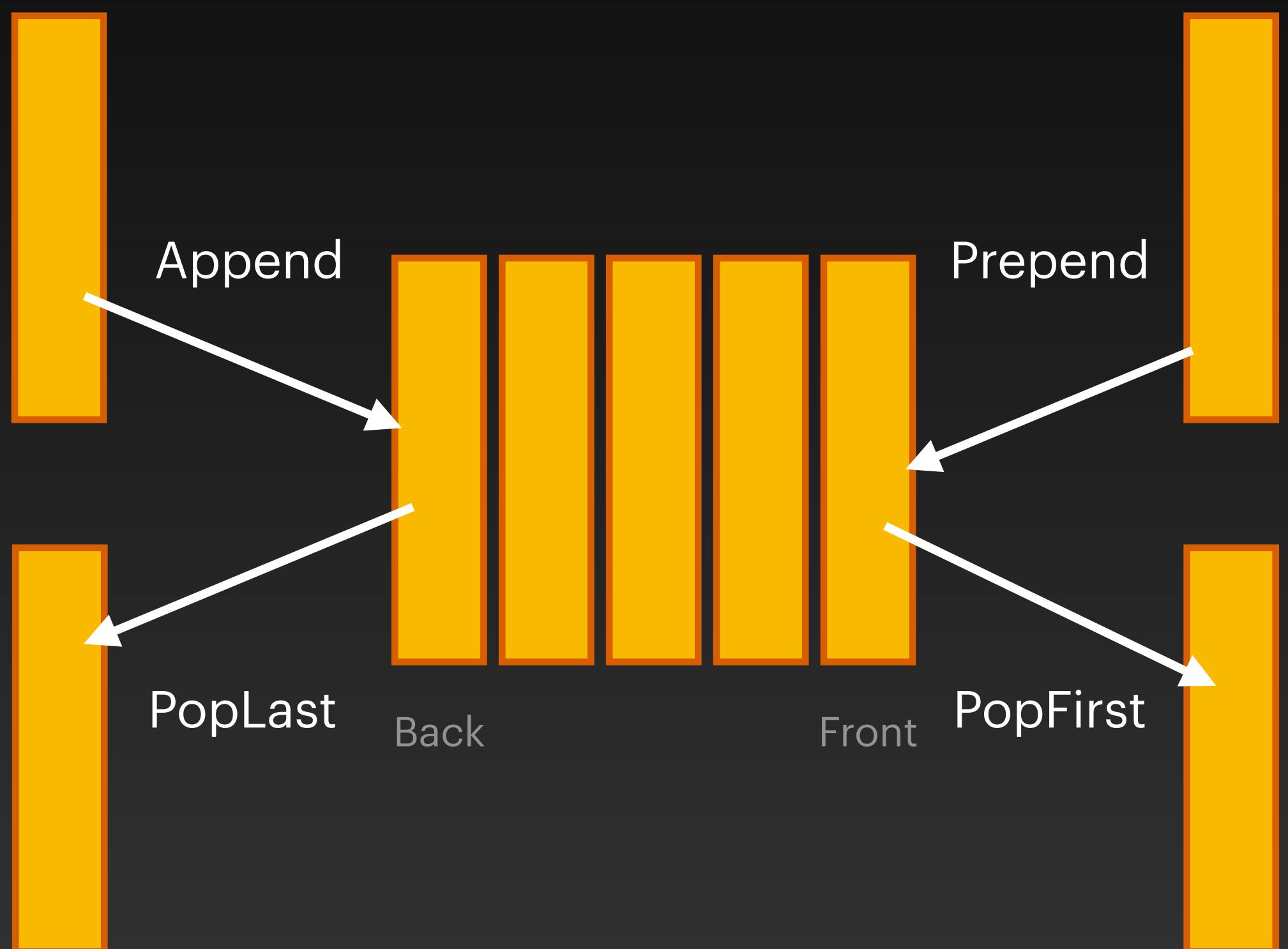
# Stacks

- Stacks are Last In First Out (LIFO).



# Deque (pronounced DECK)

- Deque<Element> stands for a “Double-Ended” Queue.
- Deques can be FIFO or LIFO.



# Deque (pronounced DECK)

```
var colors: Deque = ["red", "yellow", "blue"]
colors.replaceSubrange(1...2, with: [])
// colors: ["red"]
```

```
colors[0] = "purple"
// colors: ["purple"]
```

# Deque: Complexity

- prepend:  $O(1)$  instead of  $O(N)$
- popFirst:  $O(1)$  instead of  $O(N)$

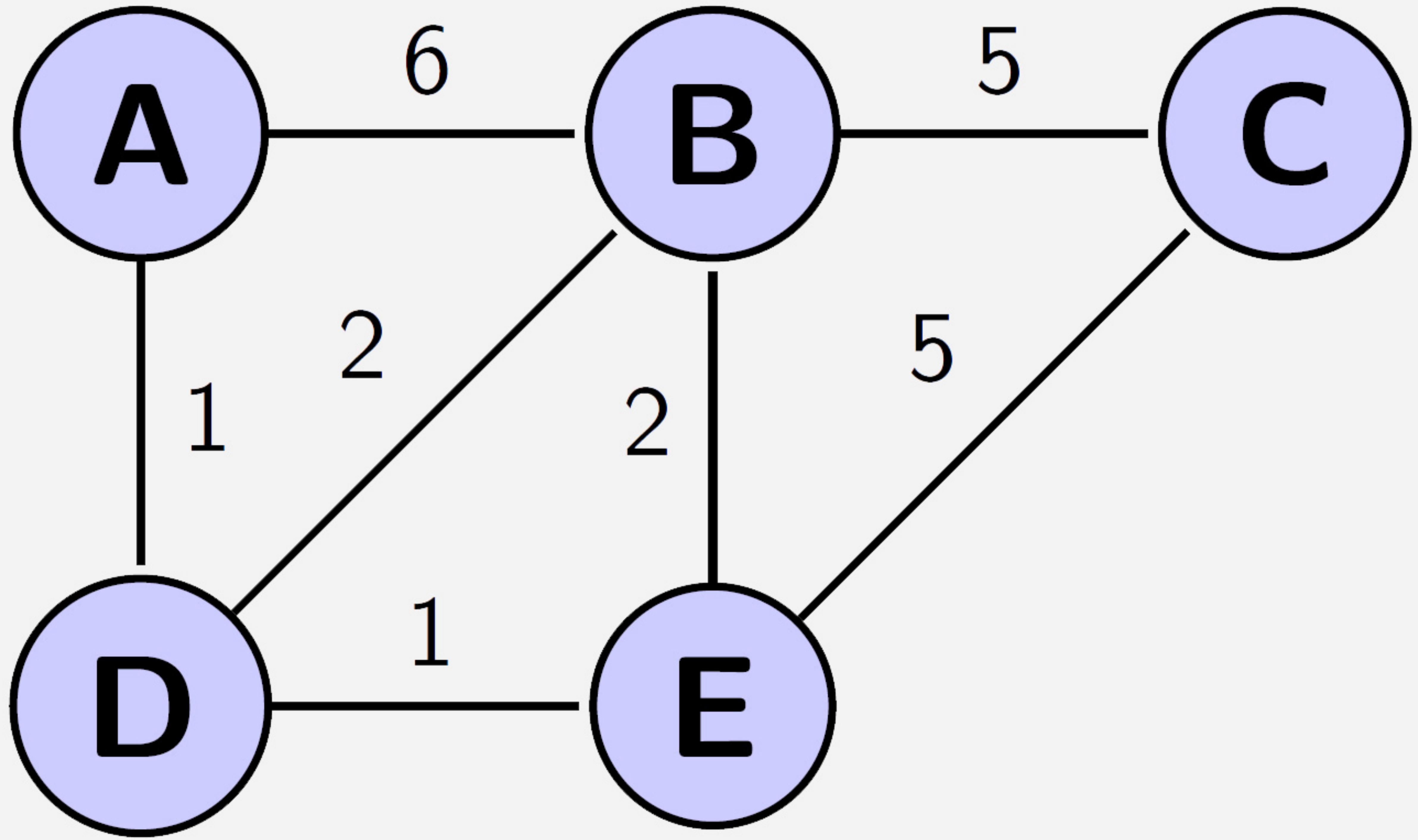


# Deque: Applications

- Can be used to check for palindromes.
- Better than Arrays
- We'll use Deques in the 0-1 BFS problem (graph-based).

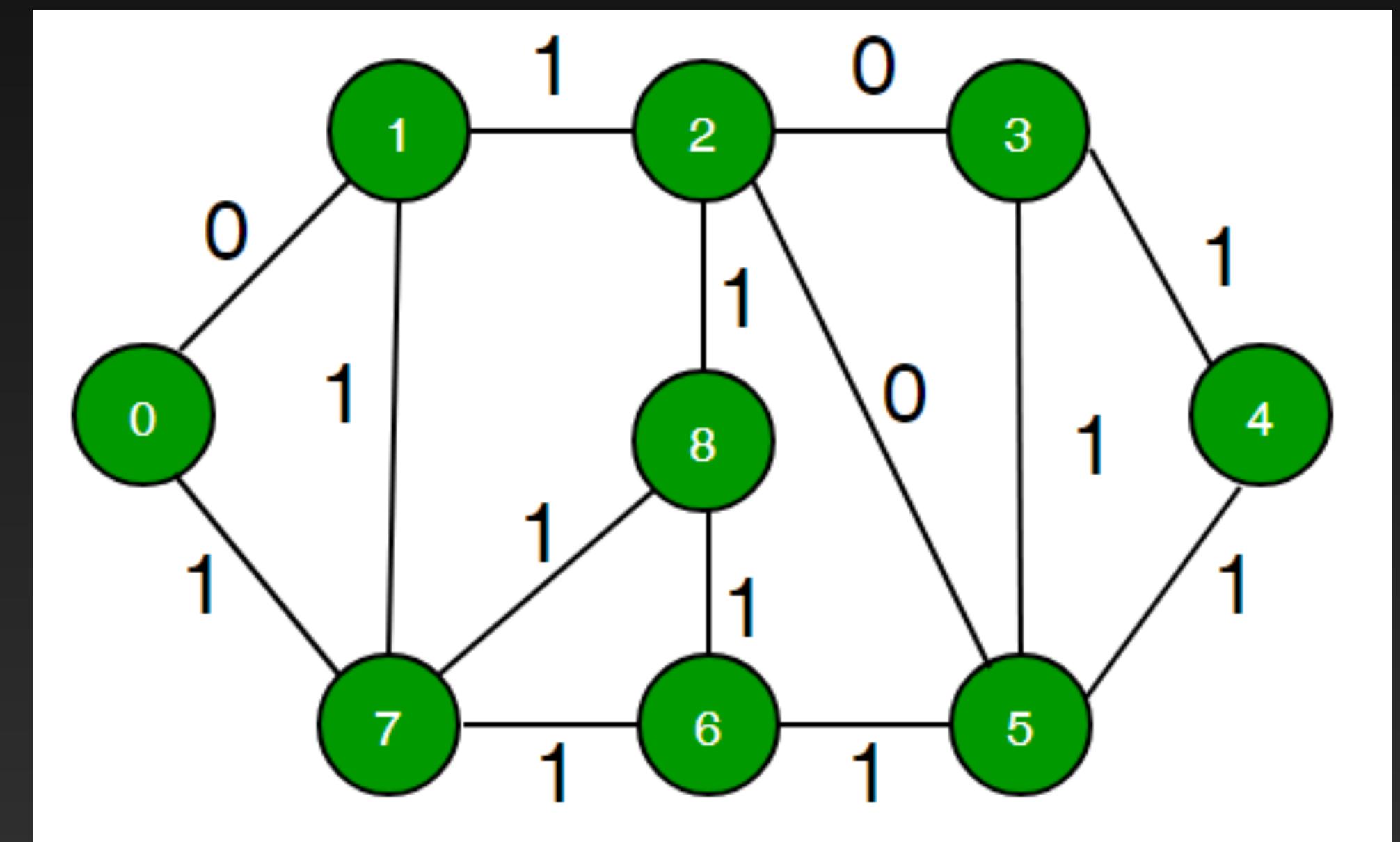
3  
1  
0  
3 1 0 2 0 1 3  
0  
1  
3

# So, what is a Graph?



# Deque: 0-1 BFS problem

- Given a graph where every edge has weight as either 0 or 1. A source vertex is also given in the graph. Find the shortest path from source vertex to every other vertex.



# Deque: 0-1 BFS solution

1. How do we represent the graph?
2. Do we need multiple traversals?
3. Code up the base BFS / DFS
4. Add in BFS modifications / extensions

# Deque: 0-1 BFS solution

## How do we represent the graph?

```
// Represents edges in the graph
class Edge {
    var to: Int
    var weight: Int

    init(to: Int, weight: Int) {
        self.to = to
        self.weight = weight
    }
}
```

```
// Store edges as an adjacency list.
var edges: [[Edge]] = []
```

# Deque: 0-1 BFS solution

## Do we need multiple traversals?

- We want to traverse the graph from a single start vertex.

# Deque: 0-1 BFS solution

## Code up the base BFS

```
import Collections
```

```
// double ended queue to do BFS.
```

```
var Q: Deque = [src]
```

```
dist[src] = 0
```

```
while !Q.isEmpty {
```

```
    let v = Q[0]
```

```
    _ = Q.popFirst() // Optimisation
```

```
// Traverse BFS
```

```
for i in 0 ..< edges[v].count { ...
```

# Deque: 0-1 BFS solution

## Add in BFS modifications / extensions

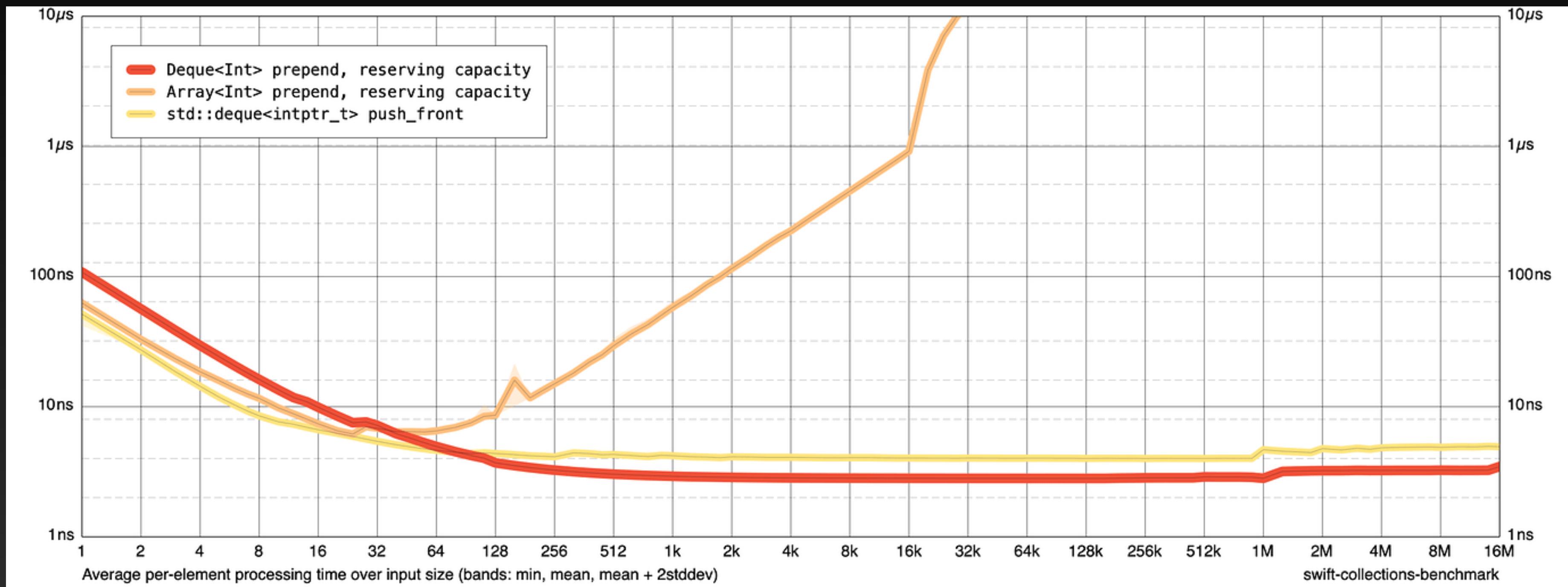
```
if edges[v][i].weight == 0 {  
    Q.prepend(edges[v][i].to) // Optimisation  
} else {  
    Q.append(edges[v][i].to)  
}
```

# 0-1 BFS Complexity

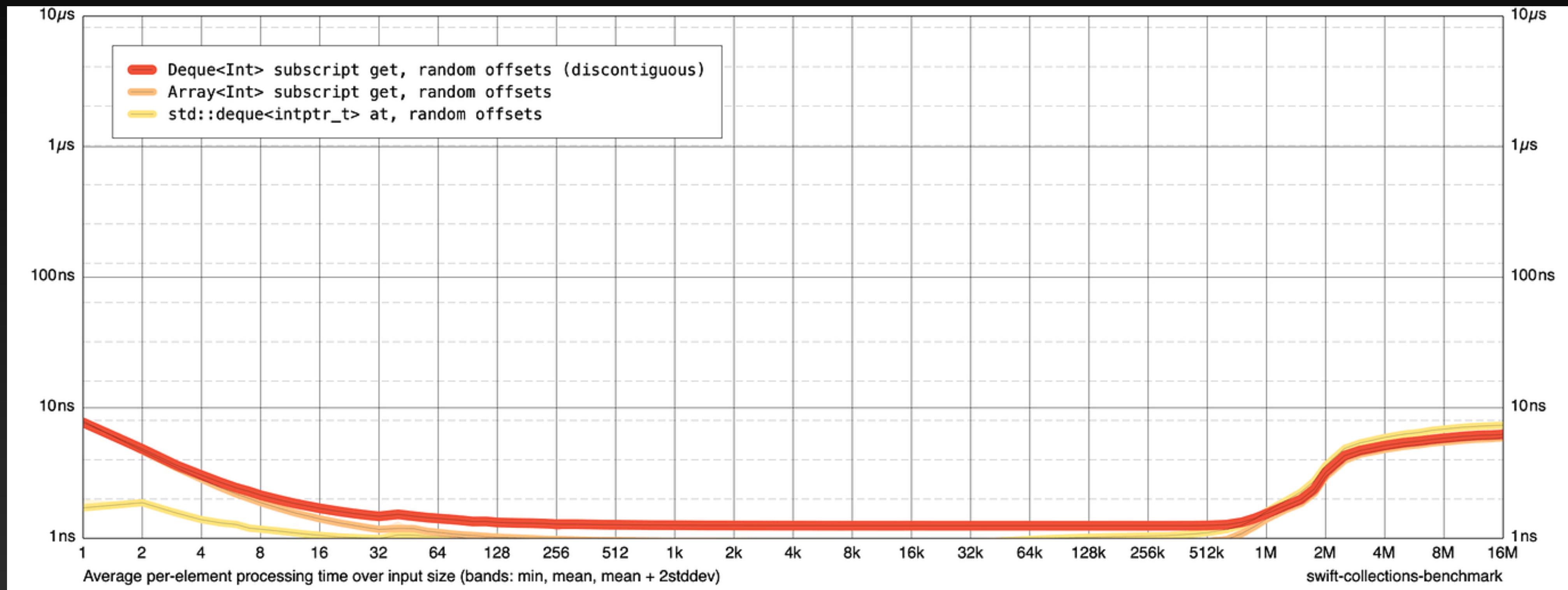
- With Deques, we have  $O(V+E)$  runtime.



# Deque: Performance Profiles



# Deque: Performance Profiles



# OrderedSet / OrderedDictionary

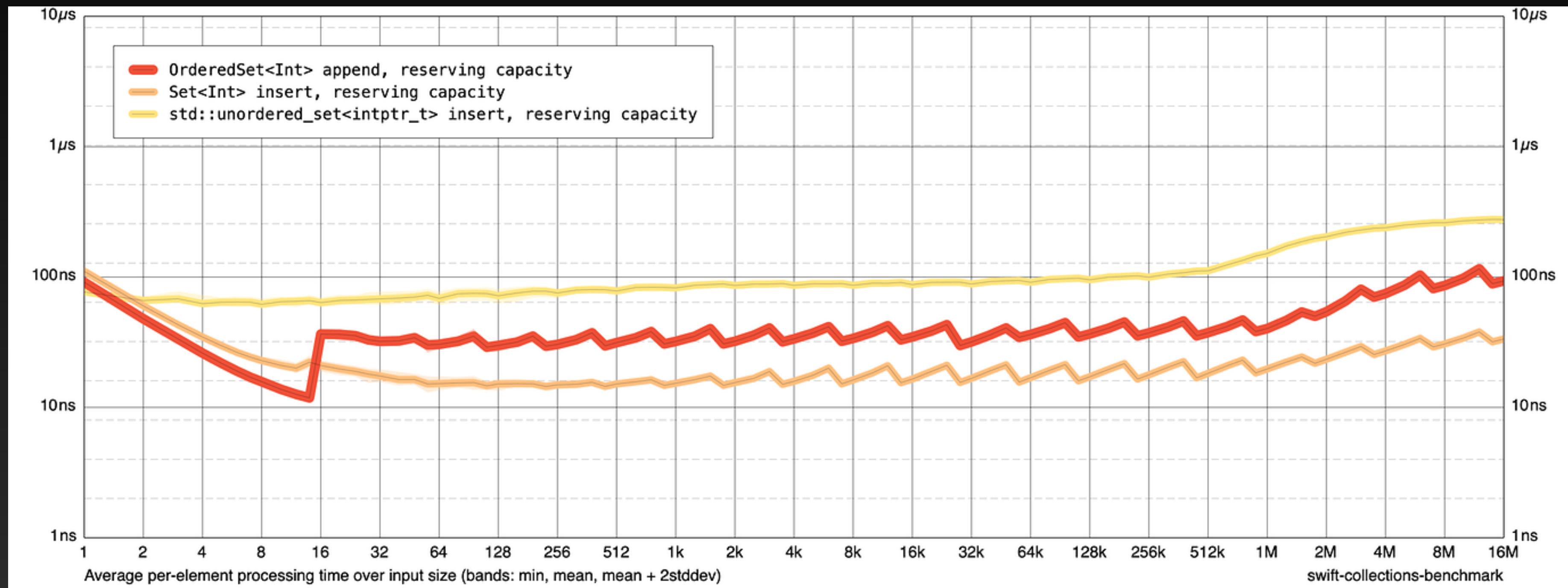
- OrderedSets are the same as Sets i.e. all elements are unique, except they maintain an ordering.
- OrderedDictionaries are ordered collections of key-value pairs.
- They support random access traversal.



# OrderedSet code example

```
let a: OrderedSet = [1, 2, 3, 4]
var b: OrderedSet = [4, 3, 2, 1]
a == b // false
b.sort() // `b` now has value
[1, 2, 3, 4]
a == b // true
```

# OrderedSet: Performance Profile



# Summary

- Why Swift Collections?
- What is Swift Collections?
- Introduction to Queues and Stacks
- Deque in a real problem.
- OrderedSet and OrderedDictionary overview.





A dark night sky filled with stars and a bright green aurora borealis over a mountain range.

**Thank You!!**

Twitter: @spidey\_vitamins

GitHub: @LemonSpike

Pranav Kasetti 6/7/22