Linked List & Stacks & Queues

Kevin Yu

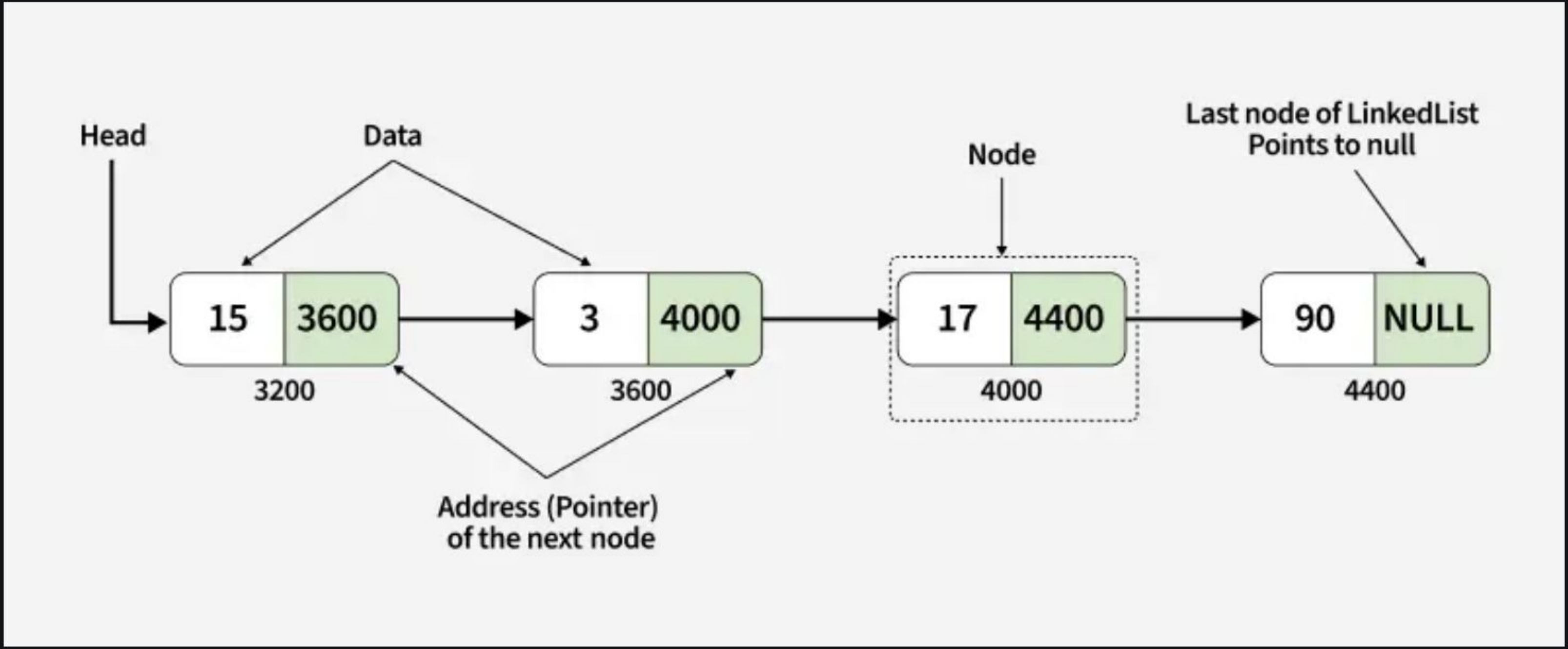
Linked List

The individual items are called nodes and connected with each other using links.

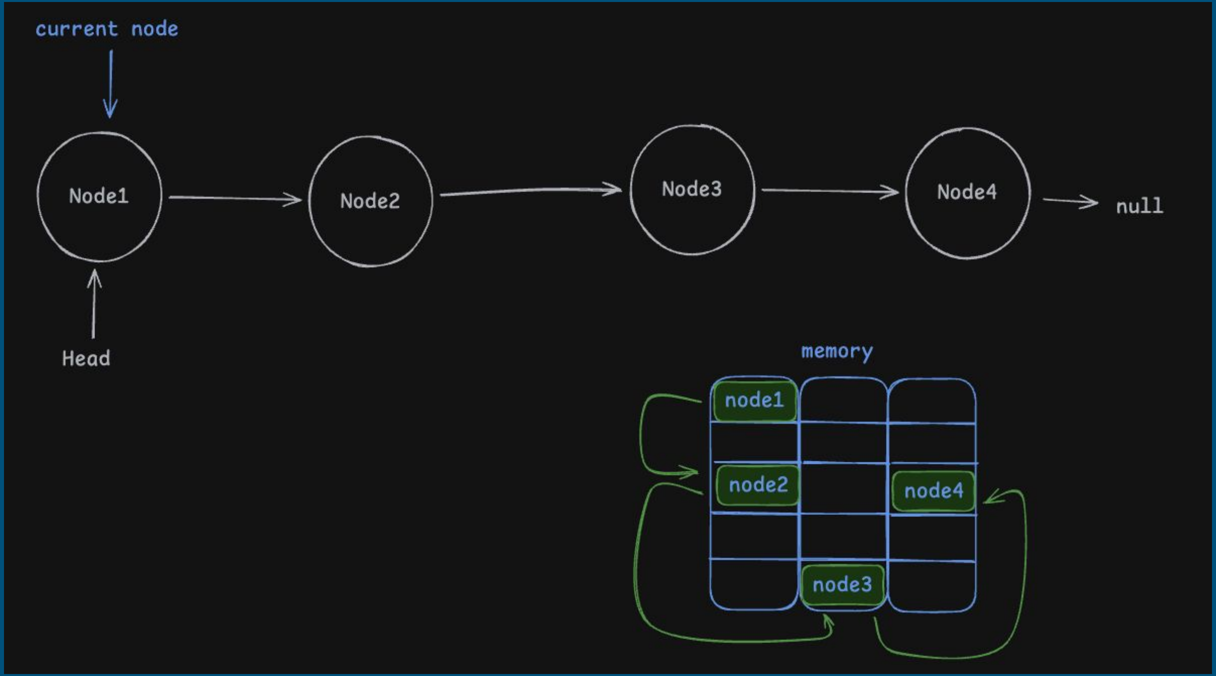
* A node contains two things first is data and second is a link that connects it with another node.

* The first node is called the head node and we can traverse the whole list using this head and next links.

Linked List

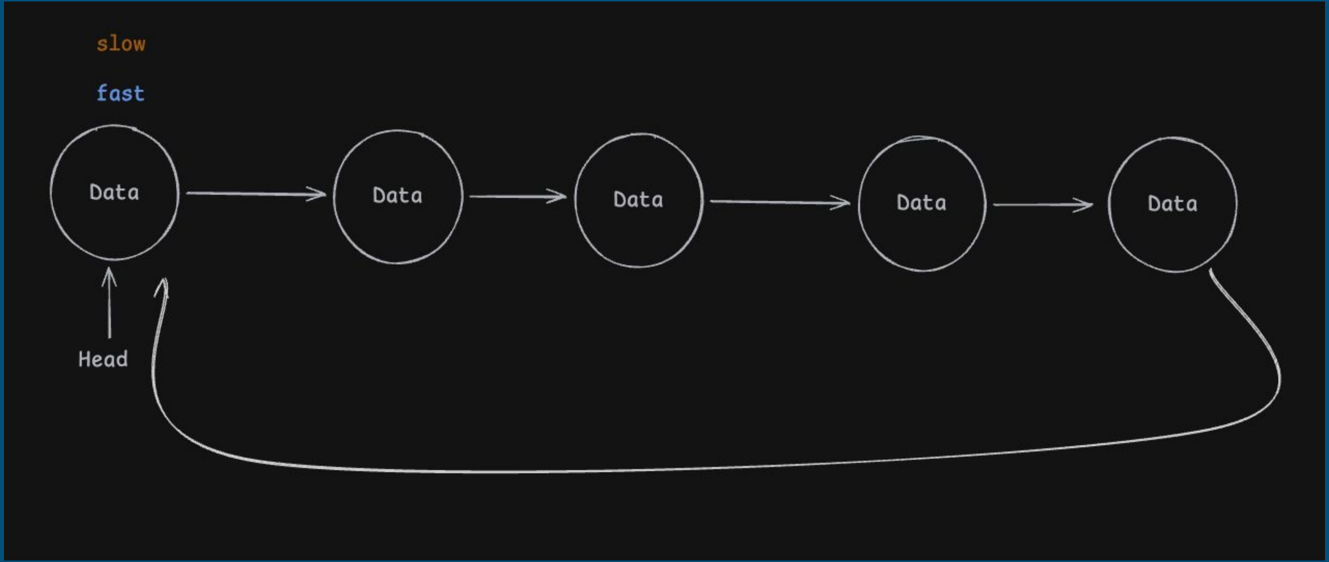

Head
Data
Node
Last node of LinkedList Points to null
15
3600
3200
3
4000
3600
17
4400
4000
90
NULL
4400
Address (Pointer) of the next node

Linked List - Memory Visualization



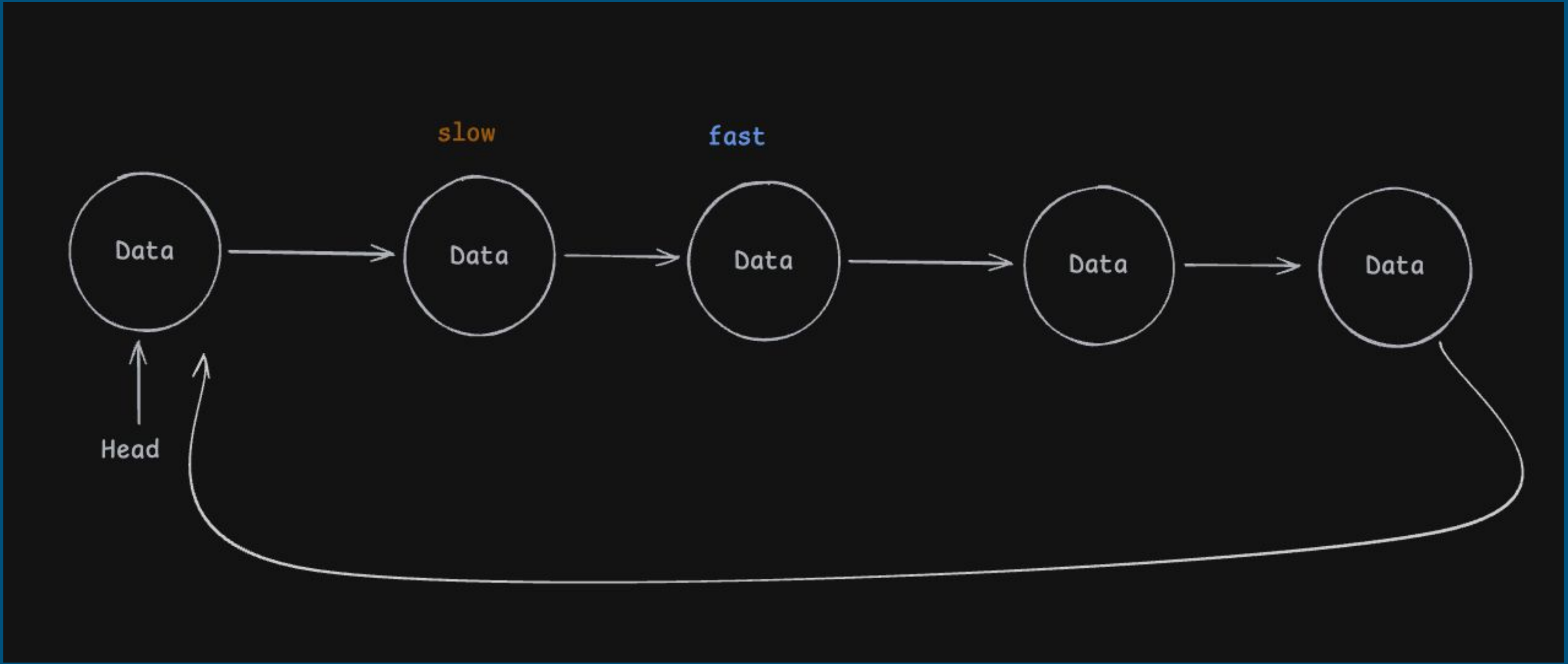
Two Pointers (Slow and Fast Pointers) Part 1

https://leetcode.com/problems/linked-list-cycle/



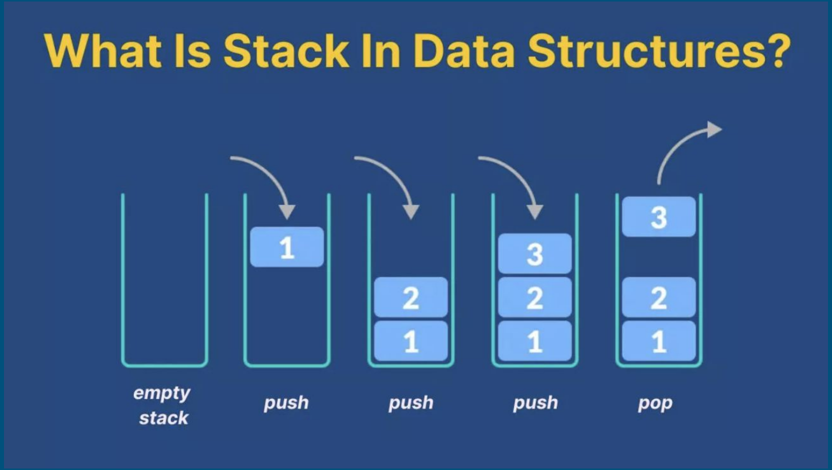
Two Pointers (Slow and Fast Pointers) Part 2

https://leetcode.com/problems/linked-list-cycle/

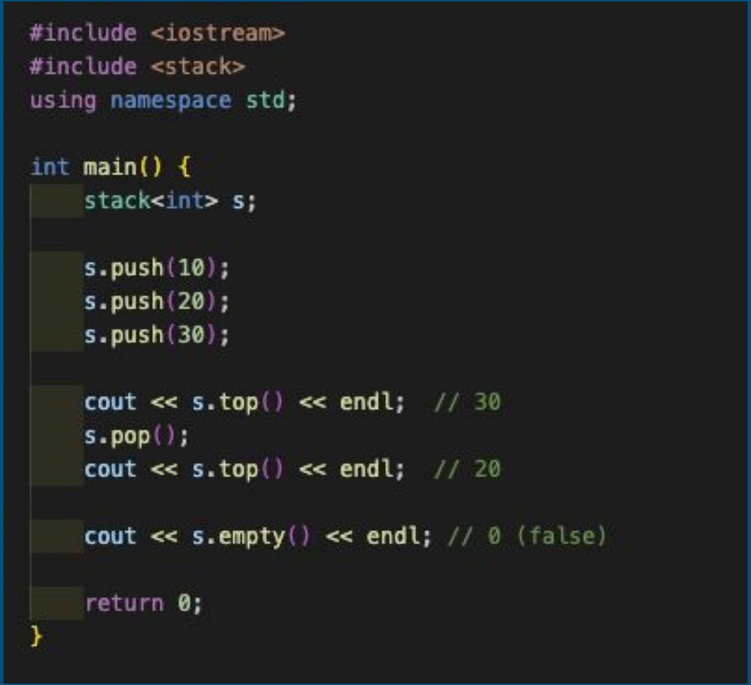


Stacks

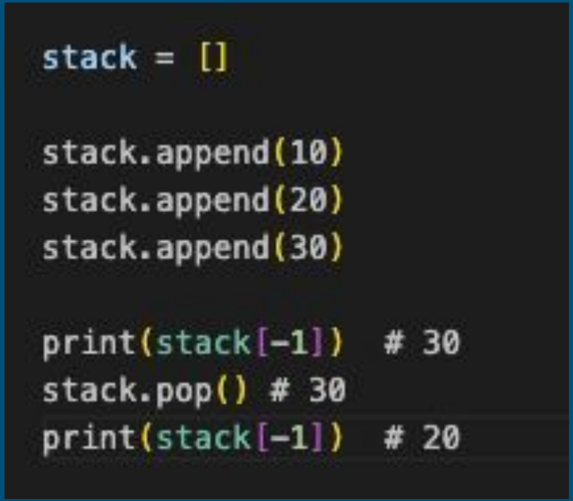
A Stack is a linear data structure that follows the LIFO(Last In First Out) order: elements that are inserted last, comes out first. Stacks can be implemented with dynamic arrays (vectors)



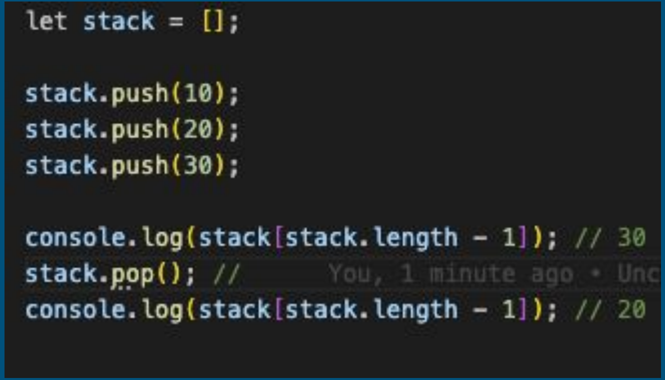
Examples of Stacks



```cpp
#include <iostream>
#include <stack>
using namespace std;

int main() {
    stack<int> s;

    s.push(10);
    s.push(20);
    s.push(30);

    cout << s.top() << endl;  // 30
    s.pop();
    cout << s.top() << endl;  // 20

    cout << s.empty() << endl; // 0 (false)

    return 0;
}
```



```python
stack = []

stack.append(10)
stack.append(20)
stack.append(30)

print(stack[-1])  # 30
stack.pop() # 30
print(stack[-1])  # 20
```



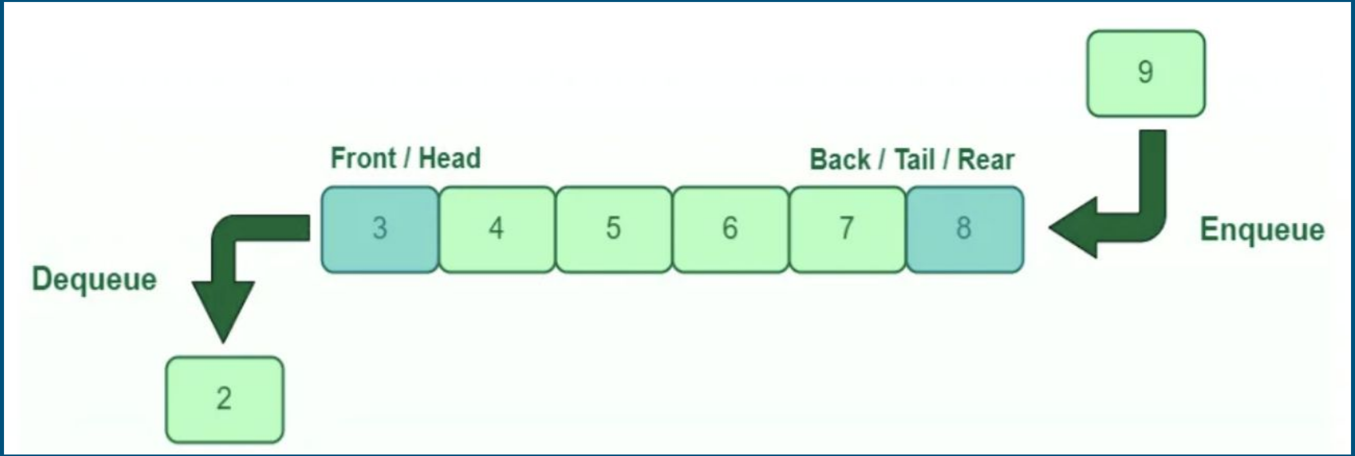
```javascript
let stack = [];

stack.push(10);
stack.push(20);
stack.push(30);

console.log(stack[stack.length - 1]); // 30
stack.pop(); //       You, 1 minute ago • Unc
console.log(stack[stack.length - 1]); // 20
```

Example Problem -

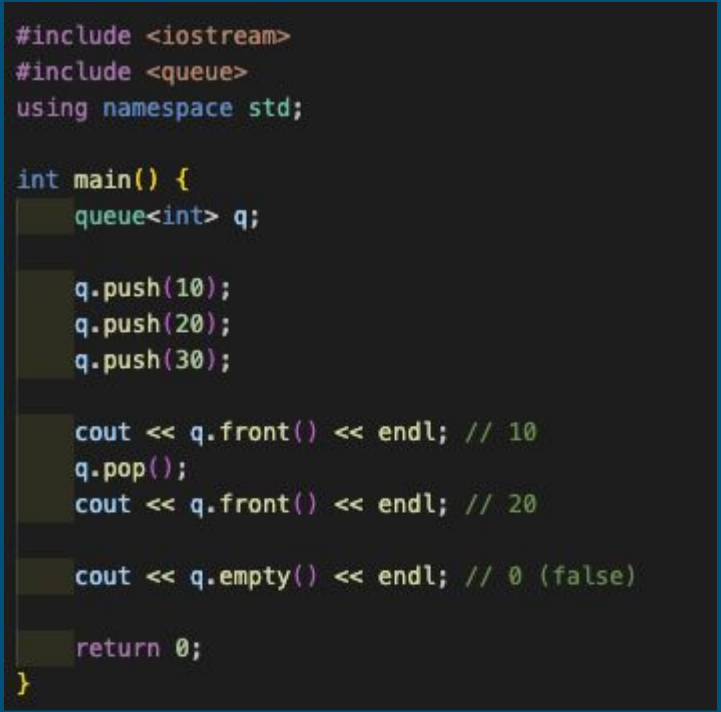
https://leetcode.com/problems/valid-parentheses/

Queues

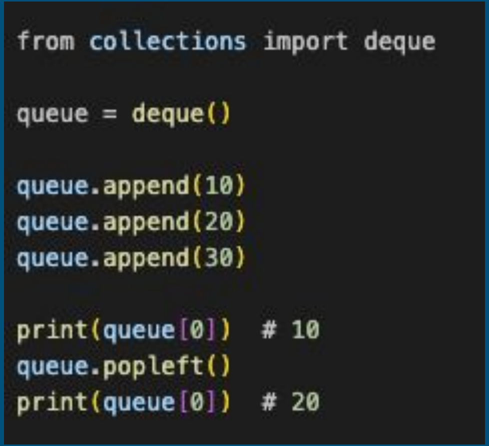
A Queue is a linear data structure that follows the FIFO(First In First Out) order: elements that are inserted FIRST, comes out FIRST. Queues can be implemented using a linked list.



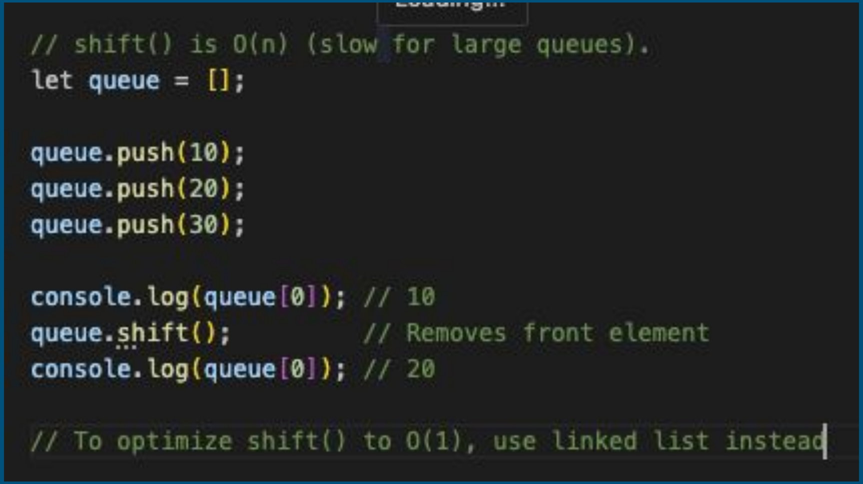
Examples of Queues



```cpp
#include <iostream>
#include <queue>
using namespace std;

int main() {
    queue<int> q;

    q.push(10);
    q.push(20);
    q.push(30);

    cout << q.front() << endl; // 10
    q.pop();
    cout << q.front() << endl; // 20

    cout << q.empty() << endl; // 0 (false)

    return 0;
}
```



```python
from collections import deque

queue = deque()

queue.append(10)
queue.append(20)
queue.append(30)

print(queue[0])  # 10
queue.popleft()
print(queue[0])  # 20
```



```javascript
// shift() is O(n) (slow for large queues).
let queue = [];

queue.push(10);
queue.push(20);
queue.push(30);

console.log(queue[0]); // 10
queue.shift();         // Removes front element
console.log(queue[0]); // 20

// To optimize shift() to O(1), use linked list instead
```

Thank You