Agenda

Intros:

- WWCode @ Code Fellows PDX
- Link to slides: https://github.com/wwcodeportland/study-nights/tree/master/algorithms

Data Structure Summary:

- Queues - How and Why

Lab Time:

- Pair Programming + 3 Queue Algorithms

Algorithms Study Night

WOMEN WHO CODE PORTLAND

Code Fellows PDX

Leadership Team

Caterina
Director

Richa
Skills Development Lead

Shiyuan
Design Lead

Tricia
DevOps Lead

Sabina
Events Lead

Sarah Joy
JavaScript Lead

Keeley
Community Lead

Alia
Algorithms Lead

Upcoming Events - June

Featured:

- 3rd Anniversary Celebration @ New Relic – Tue, June 12th, 5:30 PM

Throughout June:

- DevOps Study Night: DevOps and Kubernetes @ Vevo
  – Wed, June 7th, 6 PM
- Roll Call: Open Source Bridge @ Eliot Center – Tue, June 20th, 8:30 AM
- Design + Product Study Night - How To Persuade Workshop @ New Relic
  – Tue, June 20th, 5:30 PM
- JavaScript Study Night @ Metal Toad – Wed, June 28th, 5:30 PM
- IoT Progressive Web Apps in Angular @ TBD – Thu, June 29th, 6 PM

{short} Code of Conduct

Women Who Code (WWCode) is dedicated to providing an empowering experience for everyone who participates in or supports our community, regardless of gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, ethnicity, age, religion, socioeconomic status, caste, or creed. Our events are intended to inspire women to excel in technology careers, and anyone who is there for this purpose is welcome. Because we value the safety and security of our members and strive to have an inclusive community, we do not tolerate harassment of members or event participants in any form. Our **Code of Conduct** applies to all events run by Women Who Code, Inc. If you would like to report an incident or contact our leadership team, please submit an **incident report form**.

Resources

WWCode @ Meetup.com

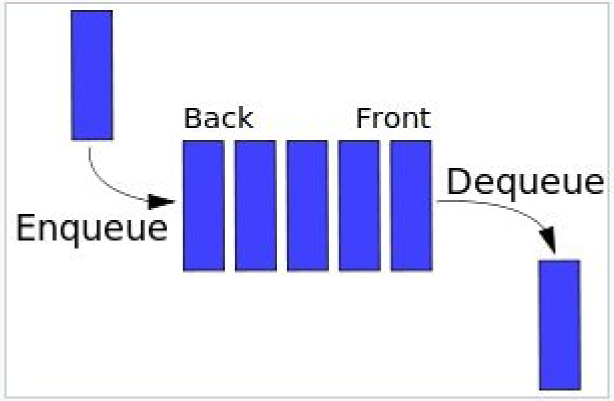
WWCode @ Slack

WWCode @ Github

Big-O CheatSheet

Basics of Queues

- First In First Out(FIFO) structure
- Most common functions include
  - Enqueue - add something to the end of the queue
  - Dequeue - remove the first element in the queue
  - Peek - look at the first element in the queue
  - isEmpty - check if the queue is empty



Representation of a FIFO (first in, first out) queue

Basic Queue Implementation

```
class Queue {
    Node first, last;
    void enqueue(Object item) {
        if (!first) {
            back = new Node(item);
            first = back;
        } else {
            back.next = new Node(item);
            back = back.next;
        }
    }
    Node dequeue(Node n) {
        if (front != null) {
            Object item = front.data;
            front = front.next;
            return item;
        }
        return null;
    }
}
```

Additional Information

Javascript Array unshift/pop

Javascript queue implementation

Java Queue Class

C++ Queue Class

Python Queue Module

Ruby Queue Class

C# Queue Class

You can find different implementations of Queues on the Wikipedia page Queue_(abstract_data_type)

3 Queue Algorithms

1 | One Task Minion

- Problem Statement

2 | Ping Pong Queue

- TopCoder

3 | Priority Queue

- TopCoder