

Stacks and Queues



Simon Robinson

Software Developer

@TechieSimon www.SimonRobinson.com



Overview



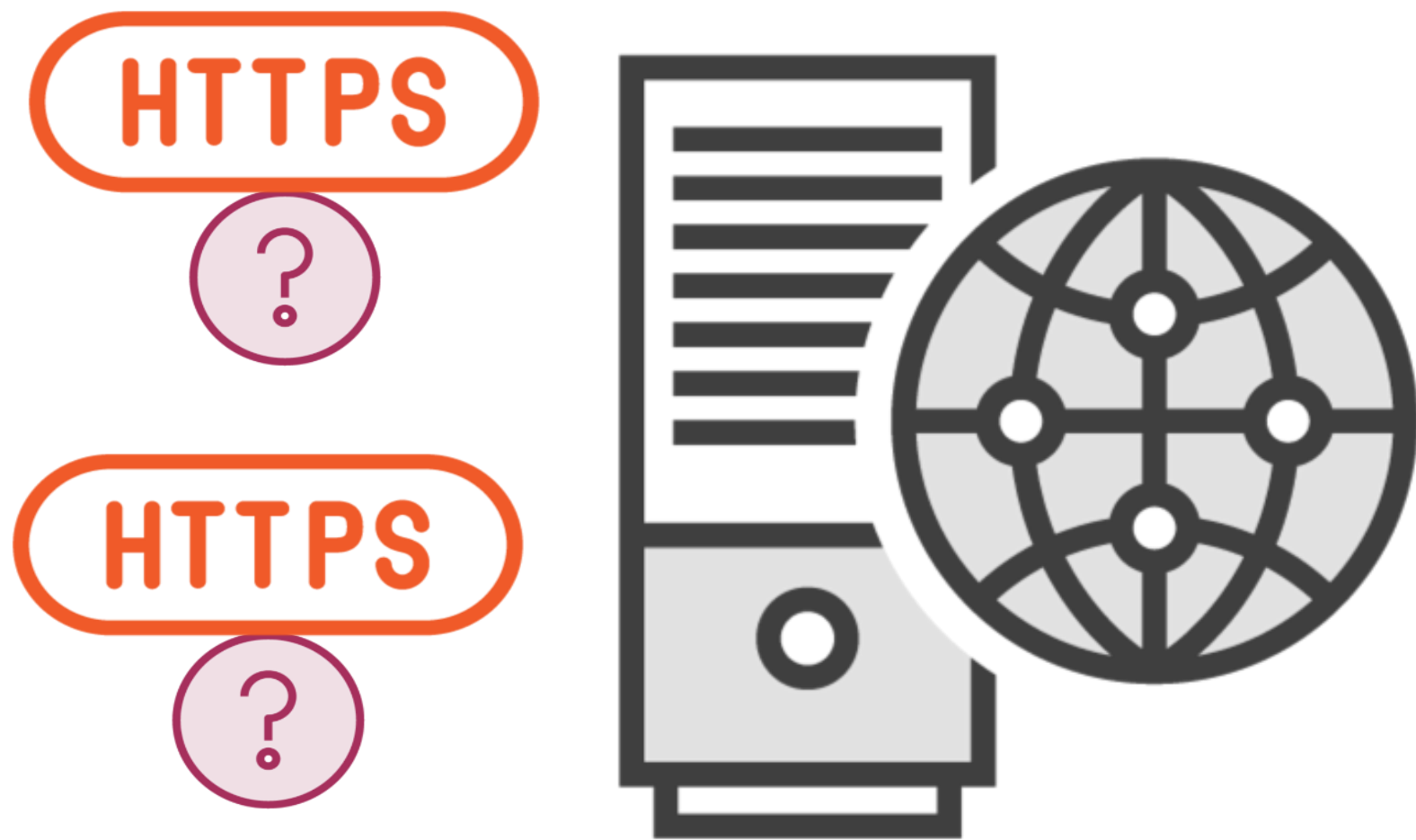
Stacks and queues

- Throughput of data
- Frequent adding/removing
- Tasks to be processed



Queues

You need `Queue<T>`



Web Page requests:

Process in order of arrival

Always choose
longest waiting item
for removal

Examples of Queues



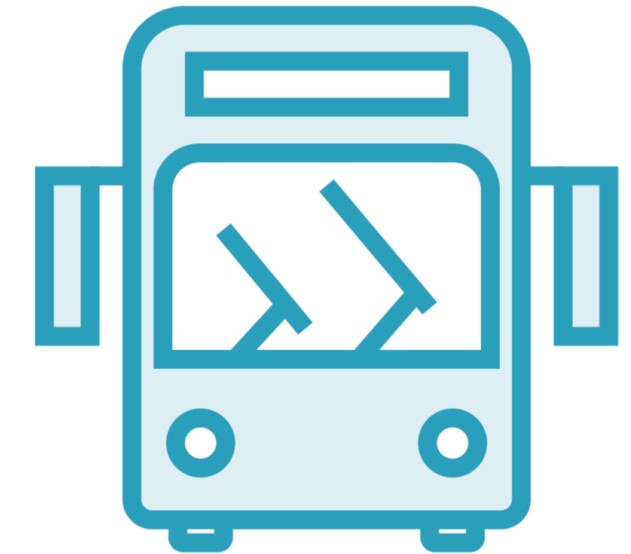
**Web Page
requests**



**Customers
to be served**



**Tasks
to be done**



**People
at bus stop**

Demo



Completely new console app

- Simulates people arriving at a bus stop
- Bus comes to pick them up
- `Queue<T>` to store list of people waiting



Code Demo

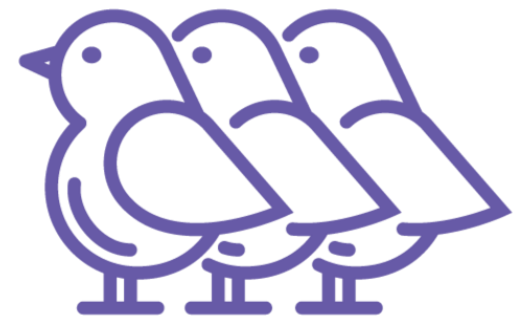
Terminology



Most
collections:

Add

Remove



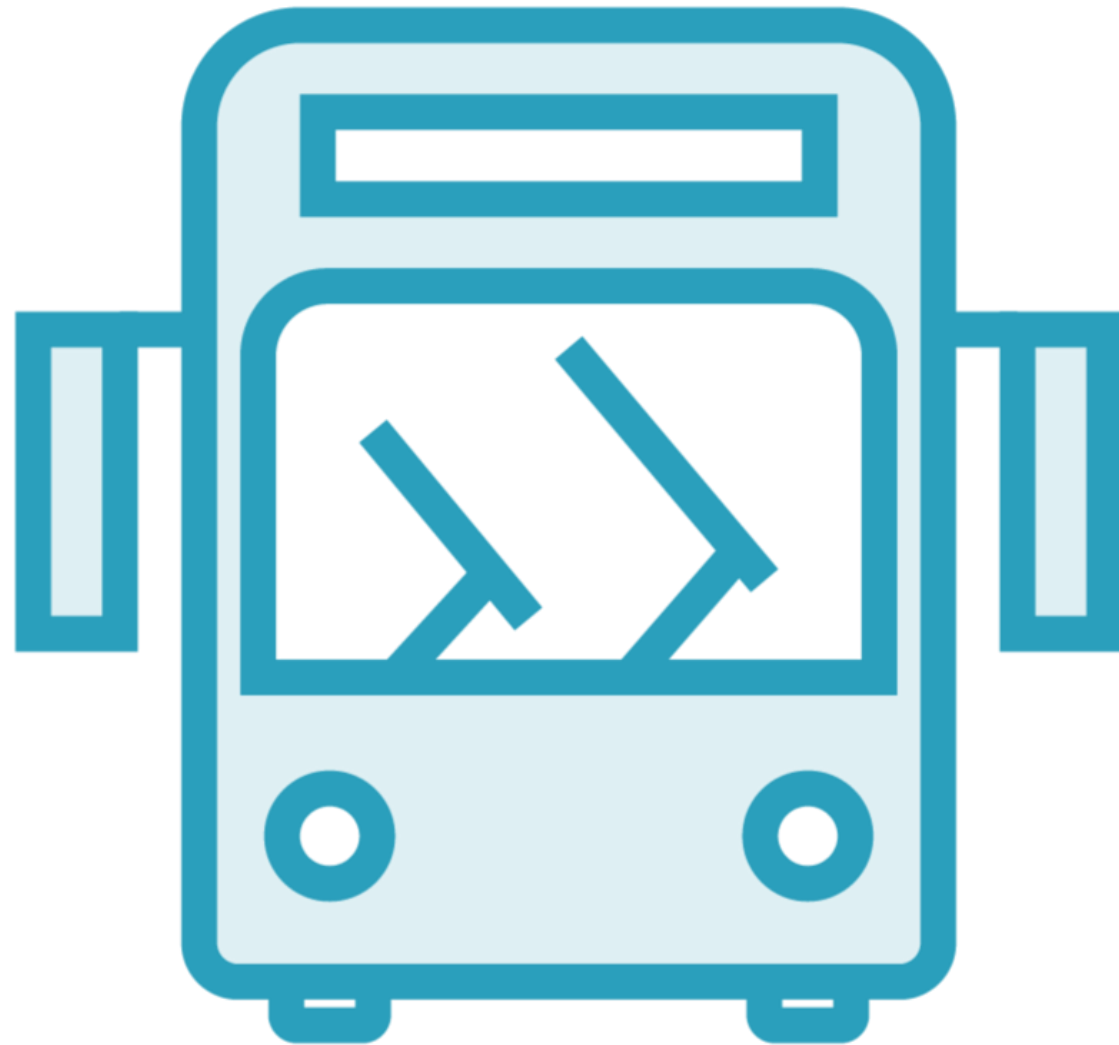
Queues:

Enqueue

Dequeue



Code Demo



Imagine bus reaches terminus, everyone gets off

- Buses are completely full
- Only passengers nearest door can get off
- People must get off in reverse order to how they got on
- Opposite order to a queue

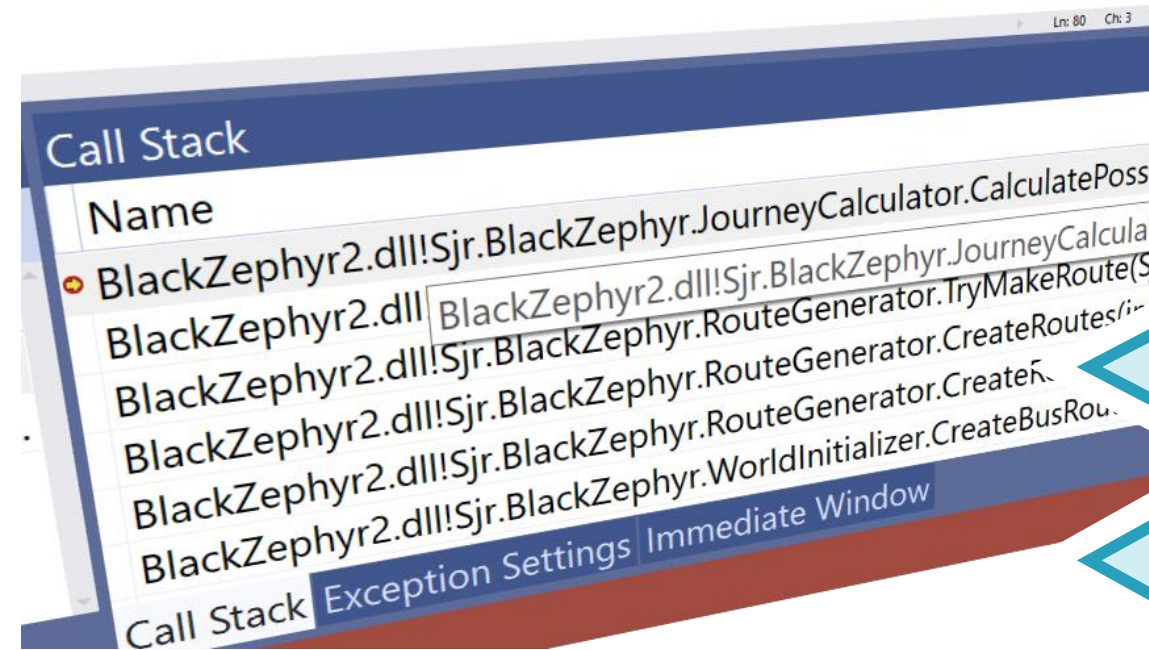
You need **Stack**<T>



Examples of Stacks



Undo button



Call stack



'S



Code Demo

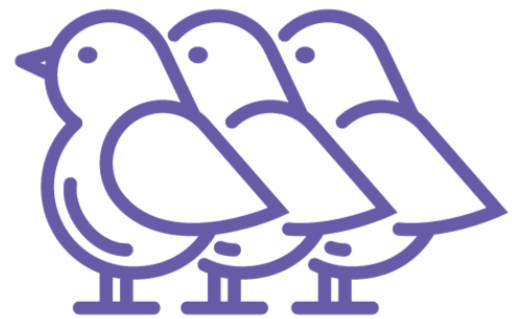
Terminology



Most
collections:

Add

Remove



Queues:

Enqueue

Dequeue



Stacks:

Push

Pop



Code Demo

Summary



Queue and stack

- For frequently adding/removing
 - Processing tasks
- Collection chooses next item
 - Queue chooses longest waiting
 - Stack chooses shortest waiting



Up next: Linked lists

