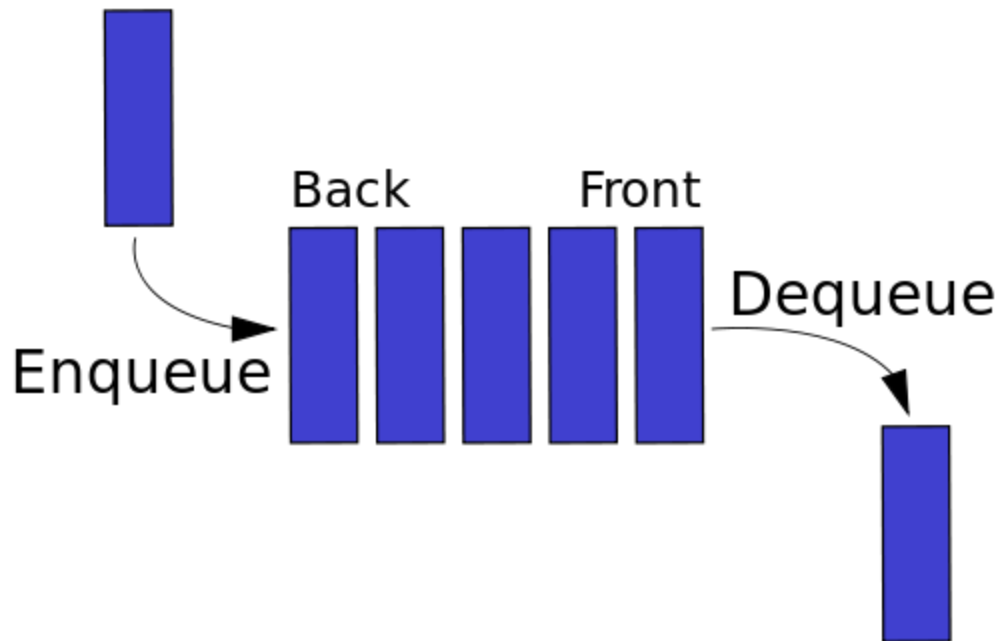


Study Write Up: Queues

Queues are an important part of computer science. They are data structures that can be thought of as a line of people; people can only enter into the line on the back and people get off the line on the front. Here is an image representing a queue:



They are the effective “opposite” of stacks, having the concept of “First-In-First-Out” (FIFO), rather than a stack’s “Last-In-First-Out” (LIFO).

One potential application of queues is in the creation of software for uploading or serving files onto the web. For example, if you have a website and you have many users, you cannot handle all requests at once. You can only handle, say, 100 at a time. A fair policy for users is first-come-first-serve. You can serve them back the requested data in order of arrival. A queue would definitely work to store the inbound requests and then send them their requests as resources are available.

Another possible use of queues is in a multitasking operating system. The CPU (central processing unit) cannot handle all jobs at once. Jobs must be grouped up and then scheduled according to a policy. A queue is a potential candidate for the job here.

Here is a simple implementation of a queue in Ruby:

```
class Queue
  attr_reader :list
```

```

def initialize
  @list = Array.new
end

def enqueue(element)
  @list << element
end

def dequeue
  @list.shift
end
end

my_queue = Queue.new

my_queue.enqueue("bananas")

my_queue.enqueue("apples")

my_queue.enqueue("kiwis")

puts "Fruits, in the order I like them: #{my_queue.list.join(", ")}"

```

This program simply adds a few items of “fruit” onto a queue and then outputs them in a string, in the order that they were added.

Queues are certainly an important part of computer science. The need for them can arise in pretty much any software development situation. They’ve have been used many times by computer scientists and programmers in all sorts of applications, and are still used by our computers everyday.