The Queue constructor should take an initial set of items

I often create a Queue and then process it with a set of concurrent workers in threads. I end up writing:

```ruby
q = Queue.new
worklist.each do |work|
  q.push work
end
```

I'd rather be able to write

```ruby
q = Queue.new(*worklist)
```

I had two choices for how to do this - Set takes an Enumerable, using each to access items, and Array takes another Array, using #to_ary if needed.
I went with the same as what Array does, because both Array and Queue are core libraries, where Set is a standard library.

#8 - 01/13/2021 06:05 AM - matz (Yukihiro Matsumoto)
I accept the idea. Take Enumerable as initial values.
Matz.

#9 - 01/29/2021 08:50 AM - ko1 (Koichi Sasada)
- Assignee set to ko1 (Koichi Sasada)

#10 - 02/12/2021 04:36 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset g1i1f0e0dfe228fd14b3f6687539ba274ba6a2d1643.

Thread::Queue.new should accept an Enumerable [Feature #17327]

Enumerable implements #to_a but not #to_array.