Overview of how I would like the strategies to function.

Heuristics:

* What are we doing
* What does it appear we are doing
* Should we specialise

What is currently going on what operation has just occurred. Are we breaking away from what we think we are doing.

What it appears to be doing is based on past actions. In this specific scenario we may have dequeued 7 times and enqueued 10 but used remove(2) twice. Is it fair to say we are functioning as a queue if items from the middle have been removed.

This should be based on current action as well as what the global goal is.

Early specialisation:

* The user can choose to specialise and lock off the code from changing.
* The user can unlock and return control to the heuristics
* The user can swap to another but allow the heuristics to choose when to swap back
* Are we actually any faster swapping to another after only a few queue operations.
* Should we “jump the gun” and swap early to hopefully make operations on a larger array quicker.

The goal is to detect what the user is doing and then adjust based on this. Take the complex thought and planning out for users that don’t want to consider it. But also allow users that wish to think and plan the ability to do this.

The goal is not ultimate efficiency but ease of use and “good enough” performance. Are we close enough where it won’t matter to use this over an ArrayList. Sequence would be good for “exploratory programming”