Direction to take the project.

Qualitative reasoning to justify the existence of the library. It make things much easier to use.

Quantitative reasoning to prove that its not much slower than traditional libraries with all these other “useful” features.

The Collections can be as easy to use and guide you through by adjusting values for you and changing its own implementation. However, for a more advanced user you can adjust certain values to fit your use cases and specific implementation.

This could be using a sequence and just defining it as per usual at the beginning then realising that it will never hold more than 10 items so restricting its initial size to that of 10 items so it never expands.

What do I want to have completed:

* 3 different strategies for sequence.
* All data structures interlink with eachother
* Proper and correct documentation for the collections library
* Benchmarks of each of my “luxury” methods against their JCF alternatives. “Sort Onwards” vs writing code to keep the ArrayList sorted.
* Each data structure can do at least “as many” things as their builtin alternatives.
* Each must implement at least two different structures from the JCF.
* Come up with heuristics of when to change between strategies for the Sequence
* Come up with strategies for the Set

Poster mentions:

* Sort onwards and its ease of use
* Performance being slightly less than ArrayList but only slightly and its sorted performance being greater.
* Builtin support for queue.
* Map being a LinkedHashMap as well as a HashMap