Data Structures and Algorithms

A Summary of Important Stuff

Author: Robby Renz

Important Points

Snapshot Iterators

- maintains its own private copy of the sequence of elements
- constructed at the time the iterator object is created
- records a "snapshot" of the sequence of elements at the time the iterator is created
- therefore, it is unaffected by any subsequent changes to the primary collection that may happen
- advantages:
  - implementing snapshot iterators is very easy
  - as it requires a simple traversal of the primary structure
- disadvantages:
  - requires *O(n)* time upon construction to copy and store a collection of n elements

Lazy Iterators

- does not make an upfront copy
- instead, it performs a piecewise traversal of the primary structure only when the `next()` method is called to request another element
- advantages:
  - typically be implemented so the iterator requires only *O(1)* construction time
- disadvantages (feature):
  - its behaviour is affected if the primary structure is modified by means other than by the iterator's own *remove* method before the iteration completes

Map ADT

- something here

Heaps

- more of something here