Approach

Cameron Burkholder

# Tuesday 09/08/2020

## Goals:

* Download all contents from Canvas, start to read through instructions
* Watch Smart Pointer video
* Read through lab description and files, look for possible places Smart Pointers would be useful
  + The array in bag would be a good place to use a shared pointer, whereas using temp variables would be good for unique pointers
* Read Chapter 4 contents related to lab (dictionaries, etc.)
* Analyze BagADT and start to brainstorm for ABag.
* Start to build ABag

## Notes

* Unique pointer: automatically deletes when it goes out of scope
* Shared pointer: uses reference counting to delete pointer when count is 0
* Weak pointer: can point to memory location without increasing reference count
* Dictionary is a data structure that will use Bag for its implementation, which will be implemented via an array that stores KeyValue objects
* Kvpair.h is already implemented, so I can start with implementing the array in Bag
* A Bag is like a stack, so I can look at the AStack implementation for starting cues
* Max dictionary size is 10, so the array can be of size 10

## Results

* I did not use a shared pointer for the array, as I could not figure out how to declare the shared pointer to the array without defining it outside the constructor
* I did not use the unique pointers either, as there wasn’t quite a place that made sense inside the bag.
* I finished implementing ABag and wrote some initial tests.
* Implementing the ABag based off of the AStack from the book was pretty straightforward, so I’m glad I did that.

# Wednesday 09/09/2020

## Goals:

* Test bag implementation
* Look at dictionary implementation in the book
* Build out Dictionary
  + Bag is built, and it looks like most of the functions in the dictionary ADT map to functions in the ABag class without much additional work
* Look for a place to use smart pointers
  + Maybe refactor ABag to use one?

## Notes

* My bag implementation appears to function properly based on the tests I wrote, so I can move on to implementing the Dictionary
* I can reuse the tests in the test file for whatever functions aren’t used in dictionary
* Based on my research, in order to use a smart pointer in the Bag implementation I would have to change the constructor so that is passes reference to a previously declared smart pointer. I may have to rely on a contrived smart pointer implementation unless I ask someone how they might implement one.
* The Key and E types passed to the dictionary reference the key and value. Combined, they represent the E type passed to Bag. For clarity, I’m going to use V for value instead of E in my implementation.
* Looks like unique\_ptr would be good for temp variables in Dictionary

## Results

* After spending some time understanding how to use smart pointers, I built out the BDictionary class referencing the book and I used smart pointers (unique\_ptr) for all my temp variables.
* After building the BDictionary class and getting it to compile, the test file runs and appears to demonstrate my class works properly.
* I still need to add testing for the unused functions of ABag in the testing file.
* I need to verify the testing file output confirms my class works

# Thursday 09/10/2020

## Goals

* Check testing file to
  + Make sure my class works as expected with the output given
  + See what functions of ABag have gone unused
* Write tests demonstrating implementation of ABag methods not yet used

## Notes

* Functions used in BDictionary
  + Insert
  + Size
  + RemoveAny
  + Find
  + Remove
  + Clear
* Functions used in ABag
  + AddItem
  + Size
  + removeTop
  + find
  + Remove
  + EmptyBag
* Unused functions in ABag
  + InspectTop
  + +=
  + bagCapacity

## Results

* I added tests for the three unused functions in the main file.