Lab 1 Approach Document

Bag Dictionary

# Assignment Objective

# Implement a dictionary using a Bag

# Assignment Requirements

Your system should have the ability to

* Use the bag ADT provided to create an array or vector - based implementation for bags
* Use your bag to implement the dictionary ADT provided you
* Incorporate one or more smart pointers
* Test your bag and dictionary implementations wit h the bagtestmain.cpp file

# Approach

* Review all files (documents, code, helps, etc.) provided as part of the assignment
* Select a data structure and implementation that could satisfy this assignment:
  + Start by building out BDictionary.h functions, then implementing ABag.h. Finishing by testing with bagtestmain.cpp.
  + Build a Dictionary on top of a
    - Bag built on top of an
    - Array (or vector) of objects 🡪 KVpairs (key/value pairs)
  + Bag fills up from bottom to top (like a stack)
    - Unlike a stack, a bag provides access to all of the contents, not just the items at the top
    - To find something in a bag you start looking from the top
  + Dictionary used for storing key/value records
    - Provides operations for Storing, Finding, and Removing records from a collection
    - Since Find is the fundamental operation of a dictionary we need a key to search on
    - For the English dictionary the key is the word itself (Key) and the definition consists of the word and its definition (Value)
    - Dictionaries use a Key/Value pair; the key provides the “comparable object” used by Find
  + Bag Operations:
    - AddItem(E) – Add item to top of array
    - Remove(E) – Remove specified item from array
    - RemoveTop(E) – Remove top item of array
    - Find(E) – Return position of item in array, starting search from top of array
    - EmptyBag() – Clear the array
    - += operator overload (adds an item) – call AddItem with operator
      * Provide additional testing in the cpp file because it is not provided by default.
    - BagCapacity() – Return size of bag
      * Provide additional testing in the cpp file because it is not provided by default.
  + Dictionary Operations:
    - Clear() – Call inherited EmptyBag()
    - Insert(E) – Call inherited
    - Remove(E) – Call inherited RemoveTop()
    - RemoveAny(E) – Call inherited Remove()
    - Find(E) – Call inherited Find()
    - Size() – Call inherited BagCapacity()

# Build Log

9/7/23 – Today I finished going over all the assignment documentation and other files and have created my approach document up to this point. I created the project folder and imported the required files from Dropbox. Tomorrow I will determine the data structure I plan to use.

9/8/23 – Today I worked with Cameron Kauffman on the implementation of the BDictionary.h functions, alongside some of the accompanying functions in ABag.h. We plan to meet tomorrow morning to implement some of the more difficult methods.

9/9/23 – Worked with Cameron on finishing the implementation of the Dictionary and Bag methods alongside smart pointers. First draft is complete with blatant errors fixed. Honing in on specific results of the functions and reviewing our work is the next step.

9/11/23 – Worked with tutor to finalize and polish functions, alongside adding additional error checking. Meeting one more time with Cameron to compare code and finalize results. Spending the rest of time today individually testing each of the functions.

9/13/23 – Created screen\_shots.docx alongside some final touch up. Submitting final project files tomorrow

9/14/23 – Final submission of project