1 Instructions

Implement a dynamic, pointer-based (i.e., not an array-based) version of the min-heap data structure in a class MinHeap. Your implementation should store key-value pairs similar to our collection class, but with slightly different functions. Thus, you won’t be inheriting from Collection. Your implementation should include public functions for:

- adding a key-value pair into the heap (insertion based on the key value)
  
  ```cpp
  void add(const K& key, const V& val);
  ```

- removing the min key-value pair (removal based on the min key), which should return false if the min heap is empty, and return true together with the removed item’s key and value if the heap is not empty
  
  ```cpp
  bool remove_min(K& the_key, V& the_val);
  ```

- providing the keys of the heap in sorted order (as a vector) via heapsort
  
  ```cpp
  void heap_sort(std::vector<K> sorted_keys) const;
  ```

- the size of the min heap in terms of the number of key-value pairs are stored
  
  ```cpp
  int size() const;
  ```

- default constructor, a copy constructor, assignment operator, and destructor

Your sort function must create a copy of the current heap structure, then repeatedly call your remove-min function to perform heap sort. Finally, you must create a set of tests to demonstrate that your heap implementation is working correctly along with an assignment writeup describing your design and any issues in your implementation. Your grade will depend in part on the tests you develop. As extra credit, this assignment is worth a full homework assignment (i.e., approximately 3% of your final grade).

2 Submission

Submit your code, test cases, and writeup to GitHub classroom (a link will be provided to basic starter-code in piazza). Be sure to follow all instructions on the homework cover sheet.