HW21: Bank Accounts: Complete Binary Tree with Heap Order Priority

Revise HW20 (or an earlier HW assignment) to have the following functionality:

1. Create an **ArrayHeap Class** which will maintain an ArrayList of Accounts as a Heap with order priority as follows:

a) data member:

```
private ArrayList<Account> arrayHeap = new ArrayList<Account>();
```

b) methods:

```
private void siftUp()
   sifts up the element at arrayHeap[size()-1]
private void siftUp(Comparator<Account> comp)
   sifts up the element at arrayHeap[size()-1] using comp
private void siftDown()
   sifts down the element at arrayHeap[0]
private void siftDown(Comparator<Account> comp)
   sifts down the element at arrayHeap[0] using comp
public boolean add(Account acct)
   Adds acct to the heap (using account number priority)
public boolean add(Account acct, Comparator<Account> comp)
   Adds acct to the heap (using account number priority) using comp
public Account removeMin()
`     removes the Account at index 0 from the heap
public Account removeMin(Comparator<Account> comp)
`     removes the Account at index 0 from the heap using comp
public boolean isEmpty()
   Tests if the heap is empty
private Account valueAt(int pos)
   Returns the Account at index pos of the heap
public Account remove(int pos)
   Removes and returns the Account at index pos of the heap
public Account remove(int index, Comparator<Account> comp)
   Removes and returns the Account at index pos of the heap using comp
public Account get(int pos)
   Returns the Account at the index pos of the heap
public int size()
   Returns the current size of the heap
public void set(int pos, Account acct)
   Sets the value of the element at index pos of the heap to acct
```

2. The **Bank** class will now maintain the database of Accounts as a **private ArrayHeap of Accounts**:

```
private ArrayHeap account;
private ArrayHeap heap;      //for creating auxiliary heaps
```

3. Modify (and add) any methods of all classes as deemed necessary.

Program testing should proceed as for the previous homework assignments with these modifications:

At initialization, a neatly formatted table of the initial database of active accounts should be printed as follows:

- **a) unsorted**
- **b) sorted by account number (using the heapsort algorithm and an auxiliary heap)**
- **c) sorted by SSN (using the heapsort algorithm)**
- **d) sorted by Name (using the heapsort algorithm)**
- **e) sorted by Balance (using the heapsort algorithm)**

At the end, before the user quits, the program prints the contents of the final database as follows:

- **a) unsorted**
- **b) sorted by account number (using the heapsort algorithm and an auxiliary heap)**
- **c) sorted by SSN (using the heapsort algorithm)**
- **d) sorted by Name (using the heapsort algorithm)**
- **e) sorted by Balance (using the heapsort algorithm)**
- **f) sorted by account number (using the heapsort algorithm and using the primary heap)**

Submission Requirements:
Create a folder on Google Drive that will contain the following:
1. The source files (i.e., *.java files) for each of the implemented Classes:
pgmHW21.java
ArrayHeap.java
Bank.java; Account.java; Depositor.java, Name.java
SavingsAccount.java; CheckingAccount.java; CDAccount.java
Check.java; TransactionTicket.java; TransactionReceipt.java;
implemented Comparator Classes
implemented Exception Classes
etc.
2. The text file containing the initial database of accounts (e.g., initAccounts.txt)
3. The test cases text file (e.g., myTestCases.txt)
4. The output text file which contains all of the required program output (e.g., pgmOutput.txt)
Then, make the folder shareable and send me a link to the folder.