LAB ASSIGNMENT LESSON 24

BINARY Search

Background:

Using the code below, modify the Store class and the Store runner class to test the Looping (Iterative) Binary Search routine, and the Recursive Binary search routine. You will need to write the two binary search methods into the Store class, while you can add and modify the code below in your runner class.

(to be included in your main method. Modify as needed)

```
   int idToFind;
   int invReturn;
   int index;
   Scanner in = new Scanner(System.in);

   System.out.println("Testing search algorithm\n");
   do{
     System.out.println();
     System.out.print("Enter Id value to search for (-1 to quit) ---> ");
     idToFind = in.nextInt();
     //index = bsearch(new Item(idToFind, 0));
     //recursive version call
     index = bsearch (new Item(idToFind, 0), 0, myStore.size()-1);
     System.out.print("Id # " + idToFind);
     if (index == -1){
       System.out.println("     No such part in stock");
     }else{
       System.out.println("     Inventory = " + myStore.get(index).getInv());
     }
   } while (idToFind >= 0);
  }
```

(Include these methods in your Store class.)

```
/**
   *  Searches the myStore ArrayList of Item Objects for the specified
   *  item object using a iterative binary search algorithm
   *
   * @param  idToSearch  Item object containing id value being searched for
   * @return             index of Item if found, -1 if not found
   */

  private int bsearchL(Item idToSearch){
    return -1;
  }

/**
   *  Searches the specified ArrayList of Item Objects for the specified
   *  id using a recursive binary search algorithm
   *
   * @param  idToSearch  Id value being search for
   * @param  first       Starting index of search range
   * @param  last        Ending index of search range
   * @return             index of Item if found, -1 if not found
   */

  private int bsearchR(Item idToSearch, int first, int last){
      return -1;
  }
```

Assignment:

1. Add the above code to your *Store.java* class.
2. Complete both of the binary search methods.
3. An example output is given below.

```
Id # 15320     Inventory = 82
Id # 196     Inventory = 60
Id # 19967     Inventory = 45
Id # 2   No such part in stock
Id # 20000   No such part in stock
```