binary search assignment

code

class Tester {

public static int searchCustomerId(int customerIds[], int customerIdToBeSearched) {

int left = 0;

int right = customerIds.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (customerIds[mid] == customerIdToBeSearched) {

return mid; // Found the customerId, return index

} else if (customerIds[mid] < customerIdToBeSearched) {

left = mid + 1; // Search in the right half

} else {

right = mid - 1; // Search in the left half

}

}

return -1; // Customer Id not found

}

public static void main(String[] args) {

int[] customerIds = { 80451, 80462, 80465, 80479, 80550, 80561, 80665, 80770 };

int customerIdToBeSearched = 80462;

int index = searchCustomerId(customerIds, customerIdToBeSearched);

if (index == -1)

System.out.println("Customer Id " + customerIdToBeSearched + " is not found!");

else

System.out.println("Customer Id " + customerIdToBeSearched + " is found at index position " + index + "!");

}

}

class Tester {

public static int searchCustomerId(int customerIds[], int customerIdToBeSearched) {

int left = 0;

int right = customerIds.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (customerIds[mid] == customerIdToBeSearched) {

return mid; // Found the customerId, return index

} else if (customerIds[mid] < customerIdToBeSearched) {

left = mid + 1; // Search in the right half

} else {

right = mid - 1; // Search in the left half

}

}

return -1; // Customer Id not found

}

public static void main(String[] args) {

int[] customerIds = { 80451, 80462, 80465, 80479, 80550, 80561, 80665, 80770 };

int customerIdToBeSearched = 80462;

int index = searchCustomerId(customerIds, customerIdToBeSearched);

if (index == -1)

System.out.println("Customer Id " + customerIdToBeSearched + " is not found!");

else

System.out.println("Customer Id " + customerIdToBeSearched + " is found at index position " + index + "!");

}

}

