University of Illinois Springfield

My Blac Steve Halder

Module 3 **Review Test Submission: Module 3 Quiz**

Review Test Submission: Module 3 Quiz

User	Steve Halder
Course	CSC 385 D: Data Structures & Algorithms (Spring 2020)
Test	Module 3 Quiz
Started	2/3/20 10:07 PM
Submitted	2/3/20 10:31 PM
Due Date	2/3/20 11:59 PM
Status	Completed
Attempt Score	8 out of 10 points
Time Elapsed	24 minutes
Instructions	This quiz contains questions covering the material from Module 3. It is open book/notes, so you may use these materials during the quiz. The questions are presented one at a time, to allow your answers to be saved as you progress through the quiz. You will be able to backtrack to previous questions if you wish.
Results Displayed	All Answers, Submitted Answers, Correct Answers, Feedback

Question 1

0 out of 1 points

Consider the Java statement:

```
int comparison = object1.compareTo(object2);
```

where object1 and object2 both implement the Comparable interface. After this statement, the value stored in the variable comparison is 1. Assuming the compareTo method is written correctly, what can you say about the relation between the objects object1 and object2?

Selected Answer: ✗ object1 and object2 are equal

Answers:

object1 is less than object2

object1 and object2 are equal

✓ object1 is greater than object2

object1 and object2 point to the same object in memory

Question 2

1 out of 1 points

The code below represents what algorithm?

```
public int search(Comparable[] objArray, Comparable searchObj)
```

```
{
    int low = 0;
    int high = objArray.length - 1;
    int mid = 0;
    while (low < high)
    {
        mid = (low + high) / 2;

        if (objArray[mid].compareTo(searchObj) < 0) // searchObj > objArray[mid]
            low = mid + 1;
        else // searchObj <= objArray[mid]
            high = mid;
    }

    if (objArray[low].compareTo(searchObj) == 0)
        return low;

    return -1;
}
```

Selected Answer: ✓ iterative binary search

Answers:
sequential search
recursive binary search
interpolation search
✓ iterative binary search

Question 3

1 out of 1 points

Which of the following search algorithms is guaranteed to return the index of a given search item in a **sorted** list of items, assuming the search item is present in the list?

Selected Answer: ✓ binary, sequential, and interpolation search

Answers:
sequential search
binary search
interpolation search
binary search and sequential search
binary search and interpolation search
sequential search and interpolation search
✓ binary, sequential, and interpolation search

Question 4

1 out of 1 points

For an unsorted list of *N* items, what is the *average* time required for a successful sequential search?

Selected Answer: ✓ O(N/2)

Answers:
O(log N)
O(log log N)
✓ O(N/2)
O(N)

Question 5

1 out of 1 points

In general, interpolation search algorithms work on:

Selected Answer:	✓ only sorted lists with keys
Answers:	only unsorted lists, with or without keys
	only sorted lists, with or without keys
	✓ only sorted lists with keys
	both unsorted and sorted lists, with or without keys

Question 6

1 out of 1 points

In general, binary search algorithms work on:

Selected Answer:	✓ only sorted lists, with or without keys
Answers:	only unsorted lists, with or without keys
	✓ only sorted lists, with or without keys
	only sorted lists with keys
	both unsorted and sorted lists, with or without keys

Question 7

3 out of 3 points

Given the following array:

```
[0, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0]
```

and the following binary search algorithm:

```
public int binarySearch(int[] array, int searchValue)
{
    int low = 0;
    int high = array.length - 1;
    int mid = 0;

    while (low <= high)
    {
        mid = (low + high) / 2;

        if (array[mid] < searchValue)
        {
            low = mid + 1;
        }
        else if (array[mid] > searchValue)
        {
            high = mid - 1;
        }
        else
        {
            return mid;
        }
    }

    return -1;
}
```

When the binary search algorithm shown above is executed with a query value of 1, how many *element* comparisons; i.e., comparisons between `searchValue` and elements in the input array, will be made before algorithm returns an index, and what index will be returned? (Although the statement, `low <= high`, is also a comparison, it is not an element comparison, so you do not need to count this one in your calculations.)

Selected Answer:	✓ 7 element comparisons performed, return index = 1
Answers:	3 element comparisons performed; return index = 1
	4 element comparisons performed; return index = 1
	3 element comparisons performed; return index = 9
	4 element comparisons performed; return index = 9
	3 element comparisons performed; return index = -1
	4 element comparisons performed; return index = -1
	✓ 7 element comparisons performed, return index = 1
	7 element comparisons performed, return index = 9
	0 element comparisons performed

Question 8

0 out of 1 points

The key sequence "1, 5, 9, 13, 17, 21, 25" is evenly distributed.

Selected Answer:	✗ False
Answers:	✓ True
	False
Response Feedback:	Determine the differences between consecutive keys. If the differences between consecutive keys are equal then the sequence is a perfectly evenly distributed sequence. If they are close to equal, with no large fluctuations in differences, then it is still considered to be evenly distributed, although not perfectly so.

Tuesday, May 12, 2020 10:59:38 AM CDT

← OK