Started on	Thursday, September 14, 2023, 11:00 AM
State	Finished
Completed on	Thursday, September 14, 2023, 11:09 AM
Time taken	8 mins 53 secs
Grade	**9.00** out of 10.00 (**90**%)

Question 1
Incorrect
0.00 points out of 1.00

Binary Search has a time complexity of O(lg n). How is such a low time complexity possible?

a. It is not a general algorithm applicable to any data.
b. It assumes something about the data, namely that the data have been sorted. ✖
c. It throws away some of the data without even looking at it.
d. All of the above.
e. None of the above.

Your answer is incorrect.
The correct answer is:
All of the above.

Question 2
Correct
1.00 points out of 1.00

What is an example of an algorithm with O(1) time complexity?

a. Looking up a known index in an array, for example A[A.length-1].
b. Finding the largest value in an already sorted array.
c. Comparing two numbers and returning the larger.
d. All of the above. ✔
e. None of the above.

Your answer is correct.
The correct answer is:
All of the above.

Question 3
Correct
1.00 points out of 1.00

What is the complexity of finding n! if an algorithm such as below is used:

```
factorial(n)
    if n = 0
        return 1
    else
        return n * factorial(n - 1)
```

a. $\Theta(n)$ ✔
b. $\Theta(n \lg n)$
c. $\Theta(n^2)$
d. $\Theta(2n)$
e. $\Theta(n!)$

Your answer is correct.
The correct answer is:
$\Theta(n)$

Question 4
Correct
1.00 points out of 1.00

Which set of time complexities is in the correct order, fastest to slowest?

a. $n \lg n$, n^2, $n!$ ✔
b. $n \log n$, n, $n!$
c. n^2, $\lg n$, n
d. $n \lg n$, $\lg n$, n
e. All of the above are in the correct order.
f. None of the above are in the correct order.

Your answer is correct.
The correct answer is:
$n \lg n$, n^2, $n!$

Question 5
Correct
1.00 points out of 1.00

Which of the following would not improve the performance of QuickSort?

a. Find the median of the array to be partitioned and use it as the pivot.
b. Find the smallest item in the array to be partitioned and use it as the pivot. ✔
c. Randomize the array to increase the probability that it is not already sorted.
d. None of the above would improve the performance of QuickSort.
e. All of these would improve the performance of QuickSort.

Your answer is correct.
The correct answer is:
Find the smallest item in the array to be partitioned and use it as the pivot.

Question 6
Correct
1.00 points out of 1.00

Which would have the fastest time complexity when sorting an array?

a. MergeSort if the array is already sorted.
b. QuickSort if the array is already sorted.
c. HeapSort if the array is already sorted.
d. InsertionSort if the array is already sorted. ✔

Your answer is correct.
The correct answer is:
InsertionSort if the array is already sorted.

Question 7
Correct
1.00 points out of 1.00

If all items in an array are the same, which sorting algorithm is fastest?

a. Mergesort
b. Quicksort
c. Heapsort
d. InsertionSort ✔
e. All are the same.

Your answer is correct.
The correct answer is:
InsertionSort

Question 8
Correct
1.00 points out of 1.00

Which algorithm has the lowest time complexity in the worst case?

a. Bubblesort
b. Quicksort
c. Mergesort ✔
d. InsertionSort
e. All are the same.

Your answer is correct.
The correct answer is:
Mergesort

Question 9
Correct
1.00 points out of 1.00

If Quicksort's partition() algorithm is run once on the array {14, 22, 87, 141, 3}, with the last item as the pivot, what is the result?

a. {14, 22, 87, 141, 3}
b. {3, 22, 87, 141, 14} ✔
c. {3, 14, 22, 87, 141}
d. {14, 22, 141, 87, 3}
e. None of the above.

Your answer is correct.
The correct answer is:
{3, 22, 87, 141, 14}

Question 10
Complete
1.00 points out of 1.00

What is the best way to sort a million numbers?

Merge Sort for its good performance and most favorable worst case time complexity.

Comment: