

Status Finished**Started** Tuesday, 4 November 2025, 9:20 AM**Completed** Tuesday, 4 November 2025, 9:50 AM**Duration** 29 mins 33 secs**Grade** 71.0 out of 100.0

Question 1

Correct

Mark 6.0 out of 6.0

Which of these statements about recursion is true?

- ☐ a. Recursion cannot solve sorting problems
- ☐ b. Recursion is always faster than loops
- ☐ c. Recursion always uses less memory than iteration (=loops)
- ☒ d. Recursion can always be replaced by iteration (=loops) ✓

The correct answer is: Recursion can always be replaced by iteration (=loops)

Question 2

Correct

Mark 6.0 out of 6.0

How do you access the last element of a Java 2-dimensional array declared as:

int[][] arr = new int[3][4];

- ☐ a. arr[12]
- ☒ b. arr[2][3] ✓
- ☐ c. arr[1][3]
- ☐ d. arr[4][3]
- ☐ e. arr[3][4]

The correct answer is: arr[2][3]

Question **3**

Correct

Mark 6.0 out of 6.0

What is the first swap that BubbleSort will perform to sort this array in an increasing order?

Array:

5 6 1 3 8

- ☒ a. Swap 6 and 1 ✓
- ☐ b. Swap 1 and 5, since 1 is the minimum of the array
- ☐ c. Swap 5 and 6

The correct answer is: Swap 6 and 1

Question **4**

Correct

Mark 6.0 out of 6.0

Which of the following is TRUE about Java arrays?

- ☒ a. All of the answers are correct ✓
- ☐ b. Arrays are objects
- ☐ c. Array size is fixed once created
- ☐ d. The first element of a Java array has index 0
- ☐ e. In Java, arrays are homogeneous data structures, all elements in an array must be of the same declared data type

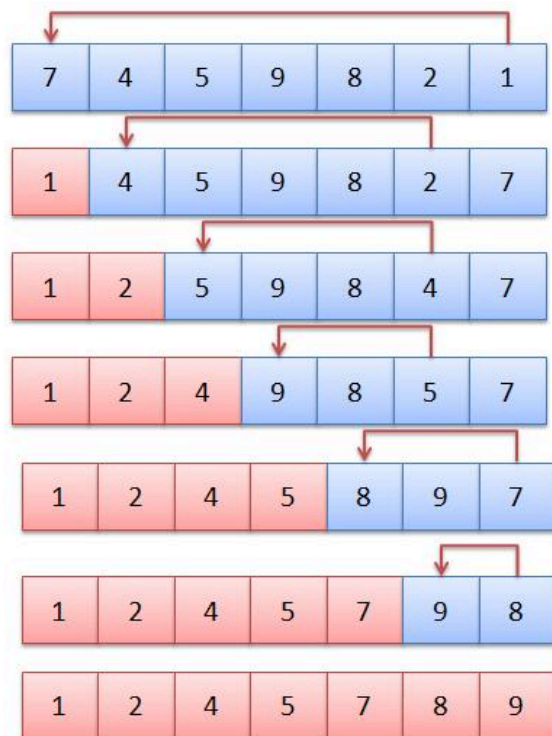
The correct answer is: All of the answers are correct

Question 5

Correct

Mark 6.0 out of 6.0

Can you recognize the sorting algorithm below?



- ☐ a. Bubble Sort
- ☐ b. Insertion Sort
- ☐ c. QuickSort
- ☒ d. Selection Sort ✓

The correct answer is: Selection Sort


Question 6

Incorrect

Mark 0.0 out of 6.0

What is the complexity of the following Java code?

```
for(int i = 0; i < n; i++) {  
    for(int j = 0; j < i*i; j++) {  
        System.out.println(j);  
    }  
}
```

- ☐ a. $O(n^4)$
- ☐ b. $O(n^3)$
- ☒ c. $O(n^2)$ 
- ☐ d. $O(n^{0.5})$
- ☐ e. $O(n)$


The correct answer is: $O(n^3)$

Question 7

Correct

Mark 6.0 out of 6.0

What happens in the “merge” step of Mergesort?

- ☐ a. Elements are swapped in-place
- ☒ b. Two sorted subarrays are combined into one sorted array 
- ☐ c. Duplicates are removed
- ☐ d. The array is partitioned around a pivot


The correct answer is: Two sorted subarrays are combined into one sorted array

Question 8

Incorrect

Mark 0.0 out of 6.0

Which sorting algorithm cannot be computed entirely in place (i.e., it requires temporary arrays for computation)?

- ☐ a. BubbleSort
- ☐ b. QuickSort
- ☐ c. MergeSort
- ☒ d. InsertionSort 
- ☐ e. NoPlaceSort

The correct answer is: MergeSort


Question 9

Correct

Mark 6.0 out of 6.0

What is the complexity of this Java code?

```
for(int i = n; i > 0; i = i / 2) {  
    System.out.println(i);  
}
```

- ☐ a. None of the above
- ☐ b. $O(n \log n)$
- ☒ c. $O(\log n)$ 
- ☐ d. $O(n)$
- ☐ e. $O(n^2)$

The correct answer is: $O(\log n)$

Question 10

Correct

Mark 6.0 out of 6.0

What is the time complexity of BubbleSort (average case) in big O notation?

- ☐ a. $O(n^{1.5})$
- ☐ b. $O(n)$
- ☒ c. $O(n^2)$ ✓
- ☐ d. $O(n^{0.5})$

The correct answer is: $O(n^2)$

Question 11

Correct

Mark 5.0 out of 5.0

Which of the following best describes how Quicksort works?

- ☐ a. Repeatedly extracts max/min
- ☐ b. Merges two sorted halves
- ☐ c. Repeatedly inserts elements into a sorted list
- ☒ d. Partitions array around a pivot element ✓

The correct answer is: Partitions array around a pivot element


Question 12

Incorrect

Mark 0.0 out of 6.0

A Java recursive function `my_print()` is shown below. What is the output of executing `my_print(3)`?

```
void my_print(int n){  
    if(n<=0) return;  
    System.out.print(n + " ");  
    my_print(n-1);  
}
```

- ☐ a. 1 3 2
- ☐ b. 3 1 2
- ☐ c. 1 2 3
- ☒ d. 3 2 1 0 
- ☐ e. 3 2 1


The correct answer is: 3 2 1

Question 13

Correct

Mark 6.0 out of 6.0

If Algorithm A has complexity $O(n^2)$ and Algorithm B has complexity $O(n)$, what is the complexity of running A and B in sequence, one after the other?

- ☐ a. $O(2n)$
- ☐ b. $O(n)$
- ☒ c. $O(n^2)$ 
- ☐ d. $O(n^3)$
- ☐ e. $O(n+n^2)$

The correct answer is: $O(n^2)$

Question 14

Correct

Mark 6.0 out of 6.0

The RandomSort algorithm is a simple algorithm for sorting an array of integers.

The idea is to shuffle the array randomly and then check whether it is sorted. If it is not sorted, you repeat the process — shuffle again until the array becomes sorted.

If n is the size of the array, what is the time complexity of RandomSort?

- ☒ a. $O(n!)$ ✓
- ☐ b. $O(n^2)$
- ☐ c. $O(n)$
- ☐ d. Since it is random, it depends
- ☐ e. $O(n \log n)$

The correct answer is: $O(n!)$

Question 15

Incorrect

Mark 0.0 out of 6.0

What is the output of this code if we execute `factorial(5)` ?

```
int factorial(int n){  
    return n*factorial(n-1); //recursive call  
    if (n<=0) return 1; //base case  
}
```

- ☐ a. 120
- ☐ b. Stack Overflow Error
- ☐ c. The factorial of 5
- ☐ d. 25
- ☒ e. Compilation Error ✗

The correct answer is: Stack Overflow Error

Question 16

Correct

Mark 6.0 out of 6.0

What causes Quicksort to reach its worst-case performance?

- ☐ a. Nothing. The performance of Quicksort is always stable, and the complexity is always $O(n \log n)$.
- ☐ b. At each step, the pivot is the median of the values in the array
- ☒ c. At each step, the pivot is always the smallest or largest element of the array ✓
- ☐ d. At each step, the pivot is randomly selected
- ☐ e. At each step, the pivot always splits the array evenly

The correct answer is: At each step, the pivot is always the smallest or largest element of the array

Question 17

Incorrect

Mark 0.0 out of 5.0

If you perform a **linear** search on a **sorted array** of 100 elements, how many comparisons will it take on average?

- ☐ a. 50
- ☒ b. 100 ✗
- ☐ c. $\log_2(100) = 6.64$, rounded up to 7
- ☐ d. 10

The correct answer is: 50