

09 Module Quiz

Started: Oct 30 at 8:09pm

Quiz Instructions

- **Purpose:** To *review and reinforce your understanding* of key concepts from the week. They are designed as a low-stakes way to assess your mastery and identify topics to revisit before exams.
- **Task:** At the end of each module, complete the quiz in Canvas. You may take each quiz as many times as you like before the deadline. Each attempt may show different questions, drawn from a larger question bank. Your highest score by the deadline will be recorded.
- **Criteria:** Full credit is earned by completing at least 80% of the weekly quizzes on time. Quizzes are unproctored and open-resource, but intended for individual review. Use them to reflect on what you've learned and where you may need more practice.



Question 1 1 pts

When is it appropriate to use Linear Search instead of Binary Search? (Select all that apply)

When you need $O(\log n)$ performance

When sorting the data would be more expensive than the number of Linear Search operations you need to do

When the data is unsorted

When the dataset is very small



Question 2 1 pts

For Binary Search to work, the data must be:

In an array (not a linked list)

All of these

Sorted

Contain only unique elements

**Question 3 1 pts**

Which sorting algorithm has the best average-case time complexity?

- Insertion Sort
- Selection Sort
- Merge Sort
- Bubble Sort

**Question 4 1 pts**

Insertion Sort has very different complexities for its best and worst cases. What determines which case occurs?

- The size of the array
- The data type being sorted
- How sorted the initial data is
- The programming language used

**Question 5 1 pts**

If an array has n elements, how many total comparisons does Selection Sort make?

- n
- n^2
- $n \log n$
- $n(n-1)/2$

**Question 6 1 pts**

Which sorting algorithm naturally lends itself to parallel processing? (That is: multiple parts of the algorithm can be run separately, at the same time.)

- Insertion Sort
- Selection Sort
- Merge Sort
- None of them

 Question 7 1 pts

You need to sort a nearly-sorted array of 1 million elements (most elements are within 10 positions of their sorted position). Which algorithm would likely perform best?

- Selection Sort
- Insertion Sort
- QuickSort
- Merge Sort

 Question 8 1 pts

Consider this array, during an Insertion Sort: [2, 4, 6, 8, 10, 3, 7, 5]. Which is the next swap that will occur?

- 3 and 10
- 8 and 5
- 3 and 4
- 5 and 7

 Question 9 1 pts

In QuickSort, if we select 2 as the pivot value in the array [3, 8, 2, 1, 9, 6], which two values will be swapped FIRST?

2 and 8

1 and 3

1 and 2

3 and 6



Question 10 1 pts

During Merge Sort, when merging two sorted subarrays [2, 7, 9] and [1, 5, 8], what are the first three elements placed in the merged array?

1, 5, 7

1, 2, 7

2, 5, 7

1, 2, 5

Quiz saved at 8:09pm

[Submit Quiz](#)