

Editorial-Solution-W3A2: Exploring Linear & Binary Search, Sorting Methods, and Complexity Trade-offs

Question 1

Which of the following best describes an algorithm?

- A) A programming language
- B) A step-by-step procedure for solving a problem
- C) A type of data structure
- D) A mathematical equation

Correct Answer: B

Explanation: An algorithm is a well-defined sequence of steps or instructions designed to solve a specific problem or perform a particular task. It provides a systematic approach to problem-solving, independent of any specific programming language.

Question 2

What is the time complexity of linear search in the worst case?

- A) $O(1)$
- B) $O(\log n)$
- C) $O(n)$
- D) $O(n^2)$

Correct Answer: C

Explanation: In the worst case scenario of linear search, the element being searched for is either the last element in the list or not present at all. This requires checking every element in the list once, resulting in a time complexity of $O(n)$, where n is the number of elements in the list.

Question 3

Which of the following is TRUE about selection sort?

- A) It's the fastest sorting algorithm for all input sizes
- B) It has a best-case time complexity of $O(n)$
- C) It's an in-place sorting algorithm
- D) It's a stable sorting algorithm

Correct Answer: C

Explanation: Selection sort is an in-place sorting algorithm, meaning it doesn't require additional memory proportional to the input size. It repeatedly selects the smallest (or largest) element from the unsorted portion and moves it to the sorted portion of the array.

Question 4

What is the primary advantage of binary search over linear search?

- A) It works on unsorted lists
- B) It has a worst-case time complexity of $O(\log n)$
- C) It's easier to implement
- D) It uses less memory

Correct Answer: B

Explanation: The main advantage of binary search is its efficiency, with a worst-case time complexity of $O(\log n)$. This makes it much faster than linear search for large datasets. However, binary search requires the list to be sorted, unlike linear search.

Question 5

In bubble sort, after the first pass, which element is guaranteed to be in its correct position?

- A) The smallest element
- B) No element is guaranteed to be in its correct position
- C) The middle element
- D) The largest element

Correct Answer: D

Explanation: In bubble sort, after the first complete pass through the array, the largest element is guaranteed to "bubble up" to the last position. This is because in each comparison, the larger element is moved towards the end of the array.

Question 6

What is the space complexity of linear search?

- A) $O(1)$
- B) $O(\log n)$
- C) $O(n)$
- D) $O(n^2)$

Correct Answer: A

Explanation: Linear search has a space complexity of $O(1)$ because it only requires a constant amount of additional memory, regardless of the input size. It doesn't create any new data structures proportional to the input size during its execution.

Question 7

Which of the following is NOT a characteristic of selection sort?

- A) It divides the input into a sorted and an unsorted region
- B) It repeatedly selects the smallest element from the unsorted region
- C) It has a time complexity of $O(n \log n)$ in all cases

Correct Answer: C

Explanation: Selection sort has a time complexity of $O(n^2)$ in all cases, not $O(n \log n)$. It divides the input into sorted and unsorted regions, and repeatedly selects the smallest element from the unsorted region to place at the end of the sorted region.

Question 8

In selection sort, after the first pass, which element is guaranteed to be in its correct position?

- A) The largest element
- B) The smallest element
- C) The middle element
- D) No element is guaranteed to be in its correct position

Correct Answer: B

Explanation: In selection sort, during each pass, the algorithm selects the smallest element from the unsorted portion of the array and places it at the beginning of the sorted portion. After the first pass, the smallest element in the entire array is guaranteed to be in its correct position at the start of the array. This process continues with each subsequent pass, gradually building the sorted portion of the array from left to right.

Question 9

What is the primary advantage of selection sort over bubble sort?

- A) Selection sort is stable
- B) Selection sort has a better average-case time complexity
- C) Selection sort performs fewer swaps
- D) Selection sort works better on partially sorted arrays

Correct Answer: C

Explanation: The main advantage of selection sort over bubble sort is that it performs fewer swaps. Selection sort makes at most $n-1$ swaps (where n is the number of elements), while bubble sort may make up to $O(n^2)$ swaps in the worst case.

Question 10

What is the primary difference between linear search and binary search?

- A) Linear search can only be used on sorted lists
- B) Binary search can be used on unsorted lists
- C) Linear search examines every element, while binary search eliminates half the remaining elements in each step
- D) Binary search is always faster than linear search, regardless of input size

Correct Answer: C

Explanation: The main difference is in their approach. Linear search examines each element sequentially until a match is found or the end is reached. Binary search, on the other hand, eliminates half of the remaining elements in each step by comparing the middle element with the target value.

Question 11

What is the best-case time complexity of bubble sort?

- A) $O(1)$
- B) $O(n)$
- C) $O(n \log n)$
- D) $O(n^2)$

Correct Answer: B

Explanation: The best-case time complexity of bubble sort is $O(n)$ when the input array is already sorted. In this case, the algorithm will make one pass through the array, comparing each adjacent pair, but no swaps will be needed.

Case Study: Optimizing Search and Sort Operations at TechMart

TechMart, a rapidly growing e-commerce company specializing in electronics, is facing challenges with its search and sort operations. As a newly hired software engineer, you've been tasked with improving their existing algorithms to enhance efficiency and customer satisfaction.

Question 12

You're implementing a search function for TechMart's customer support team to find customer orders. Which of the following Python functions correctly implements a linear search for order IDs? Linear Search should return the position.

A)

```
def search_order(order_list, target_id):  
    for i in range(len(order_list)):  
        if order_list[i] == target_id:  
            return i  
    return -1
```

B)

```
def search_order(order_list, target_id):  
    for order in order_list:  
        if order == target_id:  
            return True  
    return False
```

C)

```
def search_order(order_list, target_id):  
    for order in order_list:  
        if order == target_id:  
            return order  
    return -1
```

D)

```
def search_order(order_list, target_id):  
    return target_id in order_list
```

Correct Answer: A

Explanation: Option A correctly implements linear search. It iterates through each element of the array, comparing it with the target value. If found, it returns the index; if not found after checking all elements, it returns -1.

Question 13

TechMart is concerned about memory usage in their mobile app. In the context of sorting product lists, what does it mean for an algorithm to be "in-place"?

- A) The algorithm sorts the products without using any extra space
- B) The algorithm maintains the relative order of products with equal prices
- C) The algorithm works only on arrays of products, not on linked lists

- D) The algorithm has a time complexity of $O(n \log n)$ for sorting products

Correct Answer: A

Explanation: An "in-place" sorting algorithm sorts the elements directly in the input array, using only a constant amount of extra memory. It doesn't create a new array to store the sorted elements, thus minimizing space complexity.

Question 14

TechMart wants to implement binary search for their product catalog. What prerequisite must be met before they can use this algorithm?

- A) The product catalog must be sorted
- B) The catalog must have an odd number of products
- C) The catalog must be stored in contiguous memory locations

Correct Answer: A

Explanation: Binary search requires the list to be sorted in ascending or descending order. This is because binary search works by repeatedly dividing the search interval in half, which is only effective if the elements are in order.

Question 15

TechMart needs to implement a binary search function for their sorted product catalog. Which of the following Python functions correctly implements this search?

A)

```
def binary_search_product(catalog, target_id):
    left, right = 0, len(catalog) - 1
    while left <= right:
        mid = (left + right) // 2
        if catalog[mid] == target_id:
            return mid
        elif catalog[mid] < target_id:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

B)

```
def binary_search_product(catalog, target_id):
```

```
return catalog.index(target_id)
```

C)

```
def binary_search_product(catalog, target_id):  
    for i in range(len(catalog)):  
        if catalog[i] == target_id:  
            return i  
    return -1
```

D)

```
def binary_search_product(catalog, target_id):  
    return target_id in catalog
```

Correct Answer: A

Explanation: Option A correctly implements binary search. It maintains left and right pointers, calculates the middle index, and adjusts the search range based on the comparison with the middle element. This process continues until the element is found or the search range is