

Multiple Choice Question (MCQ) from Source Document

[BL-2: Understanding]

Question 1:

Which of the following best explains the core principle of the binary search algorithm?

Options:

- A) It sequentially checks each element from the beginning until the target is found.
- B) It repeatedly divides the sorted list in half and compares the target with the middle element to narrow the search.
- C) It requires the list to be unsorted to work efficiently.
- D) It compares the target element with the first and last elements of the list only.

Correct Answer:

It repeatedly divides the sorted list in half and compares the target with the middle element to narrow the search.

[BL-2: Understanding]

Question 2:

Why is it a mandatory precondition for a list to be sorted before applying binary search?

Options:

- A) Because calculating the middle index is impossible on an unsorted list.
- B) To ensure that the first element is always the one being searched for.
- C) So the algorithm can logically eliminate half of the remaining elements after one comparison.
- D) Because sorting the list reduces its size, making the search faster.

Correct Answer:

So the algorithm can logically eliminate half of the remaining elements after one comparison.

[BL-2: Understanding]

Question 3:

How would you contrast the approach of linear search with that of binary search?

Options:

- A) Linear search uses a sequential approach, while binary search uses a divide and conquer approach.
- B) Linear search is for large datasets, while binary search is for small datasets.
- C) Linear search requires sorted data, while binary search works on unsorted data.
- D) Linear search has a best-case complexity of $O(\log n)$, while binary search has $O(1)$.

Correct Answer:

Linear search uses a sequential approach, while binary search uses a divide and conquer approach.

[BL-2: Understanding]

Question 4:

In a binary search on a list sorted in ascending order, what is the next step if the middle element is greater than the element being searched for?

Options:

- A)** The search continues in the right half of the subarray.
- B)** The search is terminated as the element is not in the list.
- C)** The entire list is searched again from the beginning.
- D)** The search continues in the left half of the subarray.

Correct Answer:

The search continues in the left half of the subarray.

[BL-2: Understanding]

Question 5:

What is the main implication of binary search having a worst-case time complexity of $O(\log n)$?

Options:

- A)** The search time increases at the same rate as the size of the list.
- B)** The number of comparisons required grows very slowly, even for extremely large lists.
- C)** The algorithm is only efficient for small-sized datasets.
- D)** The element will always be found in the first comparison.

Correct Answer:

The number of comparisons required grows very slowly, even for extremely large lists.