Quiz Questions

Name: Yash Bhatia

Q1. What is the correct approach to calculate the middle element in binary search algorithm?

a) int mid = low + (high - low)/2;
b) int mid = low + (high + low)/2;
c) int mid = low + (high - low)/2;
d) int mid = low + ((high - low)/2)+1;

Answer: a -> int mid = low + (high - low)/2;
Explanation:
calculate the mid value and if it is not the targeted element, call the function again with 2 sub arrays, one that goes up to mid-1 and another one that starts from mid+1.

Q2. What is the worst-case complexity of binary search using recursion?

a) O(nlogn)
b) O(logn)
c) O(n)
d) O(n2)

Answer: b -> O(logn)

Explanation:
Because splitting the array in half requires constant time, the division step's time complexity is O(1). Because the size of the subproblem is half that of the original problem, the time complexity of the conquer phase is T(n/2). Because there is no need to combine anything, the combine step's time complexity is likewise O(1).

By the Master theorem, the time complexity of binary search is given by:

T(n) = a T(n/b) + f(n)

where a = 1, b = 2, and f(n) = O(1). Therefore, the case 2 of the Master theorem applies, which states that if f(n) = O(n^c) for some constant c and a <= b^c, then T(n) = O(n^c log n). Since f(n) = O(1), we have c = 0, and a = 1 <= b^c = 2^0 = 1. Therefore, T(n) = O(log n).

Q3. Which of the following is not an application of binary search?

a) To find the lower/upper bound in an ordered sequence
b) Union of intervals

c) Debugging
d) To search in unordered list

Answer:
d -> To search in unordered list
Explanation:
The condition to apply binary search in a data structure is that the data structure must be sorted.

Q4. The minimum number of comparisons for a particular record among 32 sorted records through binary search method will be:

a) 5
b) 16
c) 2
d) 8
e) 4

Answer:
a -> 5
Explanation:
Recurrence relation can be given as:
$T(n) = T(n/2) + 1$
Time complexity: $O(\log_2 n)$
So, the number of comparisons it will take is $\log_2 32 = 5$ to search for the element.

Q5. Binary Search can be categorized into which of the following?

a) Divide and conquer
b) Brute Force technique
c) Dynamic programming
d) Greedy algorithm

Answer:
a -> Divide and conquer
Explanation:
Since mid must be determined for each iteration or recursion, we divide the array in half before attempting to fix the issue.

Q6. How many comparisons does it require to find an element in a sorted array of size n using binary search algorithm?

a) $\log_2(n)$

b) $\log_2(n+1)$.
c) n
d) n/3

Answer:
a -> $\log_2(n)$.
Explanation:
The maximum number of comparisons required to find an element in a sorted array of size n using binary search algorithm is $\log_2(n)$.

Q7. Which of the following is not a variation of binary search algorithm?
a) Ternary search
b) Interpolation search
c) Linear search
d) Exponential search

Answer:
c -> Linear search.
Explanation:
Linear search is not a variation of binary search algorithm.

Q8. Which of the following is a limitation of binary search algorithm?

a) It cannot be used for finding the largest or smallest element in a sorted array
b) It can only be used for searching arrays of integers
c) It requires O(n) space complexity
d) None of the above

Answer:
a -> It cannot be used for finding the largest or smallest element in a sorted array.
Explanation:
Binary search algorithm is not suitable for finding the largest or smallest element in a sorted array, as it always starts searching from the middle.

Q9. Which of the following algorithms is similar to binary search but works on unbounded arrays?
a) Interpolation search
b) Exponential search
c) Fibonacci search
d) Linear search

Answer:
b -> Exponential search

Q10. What is the advantage of using a recursive approach over an iterative approach?

a) Consumes less memory
b) minimal code and easier to implement
c) Consumes huge memory
d) More code must be written

Answer:
b -> minimal code and easier to implement
Explanation:
A recursive approach is easier to understand as it does not contain much lines of code.

Q11. Consider an input array = [3,6,8,96,900]; key=900, what is the level of recursion?

a) 3
b) 4
c) 1
d) 6

Answer: a -> 3

Explanation:
At level 1, mid is 8.
At level 2, mid is 96 and at level 3 mid is 900 which is the key.

Q12. Which of the following is a valid scenario where binary search can be used?

a) Searching for a name in a telephone directory
b) Searching for a keyword in a document
c) Searching for a specific image on the internet
d) Searching for a word in a dictionary

Answer: d -> Searching for a word in a dictionary

Q13. What is the space complexity of the Merge Sort algorithm in the worst case?
a) O(1)
b) O(n)
c) O(log n)
d) O(n log n)

Answer:
b) O(n)

Q14. In Merge Sort, the number of comparisons needed to sort an array of n elements in the worst case is:
a) O(n)
b) O(n log n)
c) O(n^2)
d) O(log n)

Answer:
b) O(n log n)

Q15. Which of the following statements about Merge Sort is true?
a) Merge Sort can sort an array in place without using additional memory.
b) Merge Sort always has a time complexity of O(n log n), regardless of the data distribution.
c) Merge Sort is inherently unstable due to its merge step.
d) Merge Sort can achieve O(n) time complexity if the array is already partially sorted.

Answer:
d) Merge Sort can achieve O(n) time complexity if the array is already partially sorted.

Q16. In the Merge Sort algorithm, what is the purpose of using an auxiliary array during the merging step?
a) To improve cache locality
b) To reduce the space complexity
c) To avoid recursion
d) To ensure stability

Answer:
a) To improve cache locality

Q17. Which of the following sorting algorithms is based on the concept of Divide and Conquer?
a) Bubble Sort
b) Insertion Sort
c) Merge Sort
d) Radix Sort

Answer:
c) Merge Sort

Q18. What is the worst-case time complexity of Merge Sort if the merge step is implemented using a naive linear-time approach?
a) O(n)
b) O(n log n)
c) O(n^2)
d) O(log n)

Answer:
c) O(n^2)

Q19. In Merge Sort, the process of splitting an array into smaller subarrays continues until:
a) The array is completely sorted.
b) The array contains a single element.
c) The array is partitioned into two equal halves.
d) The array is reversed.

Answer:
b) The array contains a single element.

Q20. Which of the following statements about Merge Sort is true?
a) Merge Sort is not stable, as it rearranges equal elements.
b) Merge Sort always performs fewer comparisons than Quick Sort.
c) Merge Sort's worst-case time complexity can be reduced to O(n) with advanced pivot selection.
d) Merge Sort is slower than Quick Sort for all input sizes.

Answer:
a) Merge Sort is not stable, as it rearranges equal elements.

Q21. Merge Sort is an efficient sorting algorithm for:
a) Small datasets
b) Partially sorted arrays
c) Large datasets that fit in memory
d) Arrays with equal elements

Answer:
c) Large datasets that fit in memory

Q22. Which of the following is a common application of Merge Sort?
a) Cryptography
b) Graph traversal
c) Dynamic programming
d) External sorting

Answer:
d) External sorting

Q23. Which of the following statements about Quick Sort is true?
a) Quick Sort has a worst-case time complexity of O(n log n) for all inputs.
b) Quick Sort is a stable sorting algorithm.

c) Quick Sort requires additional memory for creating temporary arrays during sorting.
d) Quick Sort always performs better than Merge Sort for large datasets.

Answer: a) Quick Sort has a worst-case time complexity of O(n log n) for all inputs.

Q24. In Quick Sort, the process of rearranging elements around the pivot is known as:
a) Merging
b) Dividing
c) Partitioning
d) Combining

Answer: c) Partitioning

Q25. Which of the following pivot selection strategies aims to minimize the risk of a skewed partition?
a) Choosing the first element as the pivot
b) Choosing the last element as the pivot
c) Choosing the median of three random elements as the pivot
d) Choosing the middle element as the pivot

Answer: c) Choosing the median of three random elements as the pivot

Q26. What is the time complexity of Quick Sort in the best case scenario?
a) O(n)
b) O(n log n)
c) O(n^2)
d) O(log n)

Answer: b) O(n log n)

Q27. Which of the following is a technique to avoid worst-case behavior in Quick Sort by shuffling the input array randomly before sorting?
a) Heap Sort
b) Bubble Sort
c) Randomized Quick Sort
d) Selection Sort

Answer: c) Randomized Quick Sort

Q28. Which of the following scenarios is Quick Sort not suitable for?
a) Sorting a small array with less than 10 elements
b) Sorting an array of large integers
c) Sorting a linked list
d) Sorting an array of characters

Answer: a) Sorting a small array with less than 10 elements

Q29. The Dutch National Flag algorithm is an optimization of Quick Sort used for:
a) Sorting arrays of random integers
b) Sorting arrays of strings
c) Sorting arrays with only two distinct elements
d) Sorting arrays with duplicate elements

Answer: c) Sorting arrays with only two distinct elements

Q30. Which of the following statements about Quick Sort's in-place property is true?
a) Quick Sort requires a separate auxiliary array for sorting.
b) Quick Sort uses a secondary array for merging.
c) Quick Sort sorts the array without using additional memory.
d) Quick Sort can only be implemented with linked lists.

Answer: c) Quick Sort sorts the array without using additional memory.