1. What is a data structure in JavaScript?
   1. A variable
   2. A way to organize and store data
   3. A loop structure
   4. A function

**Explanation:** A data structure is a way to organize and store data to perform operations efficiently.

1. Which of the following is a primitive data structure in JavaScript?
   1. Stack
   2. Array
   3. Queue
   4. Linked list

**Explanation:** Arrays are considered primitive data structures in the sense that they are basic and built-in, allowing you to directly use them without needing to create your own data structures.

1. What is the time complexity of an algorithm that iterates through half of the input elements in each step?
   1. O(log n)
   2. O(n)
   3. O(n^2)
   4. O(sqrt(n))

**Explanation:** If an algorithm halves the input size in each step, it has a logarithmic time complexity, specifically O(log n).

1. An algorithm with quadratic time complexity is characterized by:
   1. Constant time
   2. Linear time
   3. Quadratic growth
   4. Exponential growth

**Explanation:** Quadratic time complexity (O(n^2)) means the running time grows proportionally to the square of the input size.

1. What is exponential time complexity known for?
   1. Efficient execution
   2. Inefficient execution
   3. Constant time
   4. Linear growth

**Explanation:** Exponential time complexity (O(2^n)) is considered inefficient as the running time grows exponentially with the input size.

1. Which of the following has a higher time complexity?
   1. O(n^2)
   2. O(2^n)
   3. Both are the same
   4. Cannot be determined

**Explanation:** O(2^n) has a higher growth rate compared to O(n^2) as the input size increases.

1. Which of the following is an example of an O(log n) time complexity algorithm?
   1. Linear Search
   2. Binary Search
   3. Both a and b
   4. None of the above

**Explanation:** An algorithm with O(log n) time complexity is typically associated with binary search. Binary search is a divide-and-conquer algorithm that efficiently finds the position of a target value within a sorted array.

1. How is space complexity different from time complexity?
   1. Space complexity focuses on memory usage, while time complexity focuses on execution time.
   2. Time complexity focuses on memory usage, while space complexity focuses on execution time.
   3. Both are the same.
   4. Neither is important.

**Explanation:** Space complexity is concerned with the memory usage, while time complexity is concerned with the execution time of an algorithm.

1. What is the purpose of analyzing space complexity?
   1. To determine the efficiency of the algorithm
   2. To minimize the number of iterations
   3. To optimize memory usage
   4. To avoid syntax errors

**Explanation:** Analyzing space complexity helps optimize the use of memory resources in an algorithm.

1. Linear search has a time complexity of:
   1. O(1)
   2. O(log n)
   3. O(n)
   4. O(n^2)

**Explanation:** Linear search has a time complexity of O(n) since it may need to iterate through the entire array.

1. What is the prerequisite for using binary search on an array?
   1. The array must be sorted.
   2. The array must be unsorted.
   3. The array must be empty.
   4. The array must be of odd length.

**Explanation:** Binary search requires the array to be sorted to efficiently locate an element.

1. Which of the following is the most efficient way of searching elements from an array?
   1. Using Binary Search Algorithm
   2. Using Linear Search Algorithm
   3. Both a and b
   4. None of the above

**Explanation:** Binary search is generally more efficient than linear search for searching elements in an array. Binary search takes advantage of the fact that the array is sorted, allowing it to repeatedly divide the search space in half, resulting in a logarithmic time complexity (O(log n)). On the other hand, linear search has a linear time complexity (O(n)), as it may need to traverse the entire array to find.

1. What is a stack in JavaScript?
   1. A data structure that follows FIFO
   2. A data structure that follows LIFO
   3. A sorting algorithm
   4. A loop structure

**Explanation:** A stack follows the Last In, First Out (LIFO) principle, where the last element added is the first one to be removed.

1. In a queue, what is the operation used to add an element to the end of the queue?
   1. Enqueue
   2. Push
   3. Add
   4. Append

**Explanation:** Enqueue is the operation used to add an element to the end of a queue.

1. In what scenario would using a stack be appropriate?
   1. Storing elements in a queue
   2. Tracking function calls and returning addresses
   3. Searching for an element in an array
   4. Sorting elements in ascending order

**Explanation:** One scenario for using a stack is to keep track of function calls and return addresses during recursion or function execution.