Data Structures and Algorithms

1. What main data structures do you know?  
2. As a data container, what are the main differences between array and list?  
3. What is the difference between singly linked list and doubly linked list data structures?  
4. What is the difference between stack and queue data structures?  
5. What is algorithm complexity?  
6. What are associative unordered and ordered containers?  
7. Explain what the binary tree is.  
8. What is the search time complexity for binary tree? Why? Is it guaranteed?  
9. What is the difference between depth-first and breadth-first searches for binary tree?  
10. What is the structure of a hash table? How does it work?  
11. What is the time complexity of add/search operations in a hash table?  
12. Explain what the binary search algorithm is.   
13. What sorting algorithms do you know?

Practical tasks  
1. How do you effectively delete sequence of elements from the middle of array?   
2. Given an array with 100 elements (numbers from 0 to 99). One element has been removed from the array. How would you find the removed element? How would you solve this if the array is sorted, or the array is not sorted?  
3. How do you find duplicates in array? How would you solve it for the array of chars?  
4. How do you find the middle element of a singly linked list in one pass?  
5. How do you detect a loop in a singly linked list?  
6. Given an array of numbers: 20, 17, 30, 21, 45, 2, 18. Form a sorted binary tree diagram. How should the sequences be changed to produce the worst binary search tree?   
7. Provide a recursive and a non-recursive solution for the tree traversal algorithms for a binary tree.

Literature

1. Algorithms (Robert Sedgewick and Kevin Wayne)  
2. Introduction to Algorithms (Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein)  
3. Programming Pearls (Jon Bentley)