DSA MCQs

Q1) What is the number of swaps required to reverse an array having N elements.

Ans1) We just need to swap the first and last element, 2nd and 2nd last element,third and third last element and so on. So the number of operations would be N/2 and if the elements are odd, the central element doesn't need a swap,so floor(N/2) would be the answer.

Q2) An array consisting of N elements indexed from 0 to N-1 is being used for the implementation of the complete-binary tree. If the index of a node is K then what would be the index of its parent. (1<=K<=N-1)

Ans 2) As the indexing is from zero to N-1, if the parent is X then the children would be 2*X+1 and 2*X+2. So the best option is floor of (K-1)/2.

Q3) What is this code trying to do.

```
 void solve(tree *root, tree *find)
{
   if (root == null)
       return;

   if (root->left == find || root->right == find || solve(root->left, find) || solve(root->right, find))
       printf(root->data)
}
```

Ans3) We are checking if the left or right node is what the argument sent or else if not the case then move to the left node or right node and print all nodes while searching for the argument node.So we would be printing the ancestors of the find node.

Q4) You are given an array consisting of 25 elements and your task is to find the min and max element of the array.

What is the minimum number of comparisons required to perform the above task.

Ans 4) `Pick 2 elements(a, b), compare them. (say a > b)`

`Update min by comparing (min, b)`

`Update max by comparing (max, a).`

Therefore, we need 3 comparisons for each 2 elements, so the total number of required comparisons will be ceil(3n)/2 – 2, because we do not need to update min or max in the very first step.

Q5) Which of the following is true about memory efficient doubly linked lists?

Ans 5) It is also called as XOR linked list.It stores the XOR of the address of the previous node and next the node.While traversing you have the address of the previous node saved in a variable and thus from the current node you can move forward as well as backward in a XOR linked list.You can read more about it from GFG.