Questions

1.Implement trie

2. Remove duplicate elements from a circular linked list and the duplicates occur consecutively in the list. It was tricky because of the edge case that the last element could be equal to the first element and since linked list is circular.

3.  Find the maximum sum from root to a leaf node of a given n-ary tree.

Basically the question was if we are given a large IP address how can we perform operation on a trie with less number of computations. I told him that instead of modifying the trie we could even modify the address by converting into a different base probably a larger base – means a shorted equivalent address.

Then he asked me the phonebook problem. How do we implement the system in our phones that maintain the contact list i.e. phone numbers and names of people?

The problem’s solution is to use a trie so that for each prefix of the string we could easily find the names and numbers associated with it.  
How do we implement the frequently called list that maintains the list of people whom we contact frequently ?

**Solution** is to use a heap and pop out the most frequently used k contacts whenever required. Heap because updation and insertion operations are fast.

1) Tell me the differences between insertion sort and quick sort. When would we use insertion sort over quick sort

Was asked to give a real world application of circular linked list.

* Round Robin scheduling technique in games.
* Audio/Video Streaming
* Circular Escalators