Whiteboarding Practice

In-class Whiteboarding Practice (1)

1. Write a method that takes an array of integers as a parameter and returns the maximum.
2. Write a method that takes a *String* s as an argument and returns a new *String* that contains all the characters in even positions from the input string. For example, for the input `String s = “Whoa”` the method would return `“Wo”`
3. Write a method that returns a String of given length n that has random characters from a - z (lowercase)

```
public String randomString (int  n) { … }
```

In-class Whiteboarding Practice (2-A)

1. **Check Permutation**: Given two strings, write a method to decide if one is a permutation of the other. What is the runtime of your solution?

In-class Whiteboarding Practice Hints (2-A)

Check Permutation. Hints for Interviewer to use:

1. Describe what it means for two strings to be a permutations of each other. Can you check the strings against that definition?
2. Come up with a brute-force solution. How long does it take?
3. There is one solution that is O(n log n). Another solution uses some space but is O(n) time (on an average)
4. Could a hash table be useful?

In-class Whiteboarding Practice (2-B)

1. **Point of intersection**: Two linked lists A (size *n*) and B (size *m*) are joined on a particular node, called the point of intersection of the linked lists. Find the point of intersection, i.e. the first node after which both lists have the same nodes.

In-class Whiteboarding Practice Hints (2-B)

Point of intersection. Hints for interviewer to use:

1. What is the brute force solution? It should take O(1) space. What is the run-time for your solution?
2. Can you come up with a faster solution?

In-class Whiteboarding Practice (3-A)

1. **Finding the boundary.** Design a method that finds the index of the first 1 from a sorted array of 0's and 1s. What is the time complexity of your solution in big-Oh notation?

In-class Whiteboarding Practice (3-B)

1. **Big Sort**. Imagine that you have a 20GB file with one string per line. How would sort the file without using more than 1GB of memory at a time?