Winter 2025

Computer Science 241

Coding Interview Problems

1 FizzBuzz

1. The following is a useless yet unfortunately common coding interview question: Write a java program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

2 Word problems!

The following are coding interview questions that pertain to the material in this course. For each of the problems below, write an **English** description of how you'd solve the problem. Specify any data structures used, and feel free to rely on algorithms, operations, etc. covered in this class. If you finish describing your approach to all of the problems, write out pseudocode for your solution.

0. **Example Problem**: Given a directed graph, design an algorithm to find out whether there is a route between two nodes.
   Example Solution: Given a source and a destination node, run a depth-first search starting from the first one. Each time a node is visited, check whether it matches the destination node and return true if so. Return false if the DFS finishes.

1. You have a text file with 1 million URLs that may contain duplicates. Determine the number of URLs that appear more than once.

2. Implement a function to check if a given binary tree is a valid binary search tree.

3. Given a sorted (increasing order) array with unique integer elements, write an algorithm to create a binary search tree with the minimum possible height.

4. Given a 1-d array of characters, remove all the continuous occurrences of characters occurring k times.

   Examples:

   - input: `[A, B, B, B, C, A, D]`
     number frequency: 2
     return: `[A, B, C, A, D]`

   - input: `[A, B, B, B, C, A, D]`
     number frequency: 3
     return: `[A, C, A, D]`

   - input: `[A, C, C, C, A, B, B, B, C, A]`
     number frequency: 3
     return: `[A, A, C, A]`

   - input: arr: `[A, C, C, C, A, B, B, A, A]`
     number frequency: 2
     return: `[A, C, A]`

   - input: arr: `[A, C, C, A, B, B, A, A]`
     number frequency: 2
     return: `[]`

5. The median of a collection of numbers is the middle value if the collection were sorted. If the size of the collection is even, then it's the average of the two middle numbers.

   Suppose you are processing a stream of numbers, and you need to be able to efficiently find the median at any point in time. Design a data structure with two operations: `addNumber`, which adds a number to the collection, and `findMedian` which returns the median of the current collection. The `findMedian` operation will be called very frequently, so it needs to be as efficient as possible.

6. You are given a list of projects and a list of dependencies (which is a list of pairs of projects, where the second project is dependent on the first project). All of a project's dependencies must be built before the project is. Find a build order that will allow the projects to be built. If there's no valid order, return an error.

   Example:

```
Input:
  projects: a, b, c, d, e, f
  dependencies: (a, d), (f, b), (b, d), (f, a), (d, c)
  Output: f, e, a, b, d, c
```