Backend Engineer Intern Take Home Test

The purpose of this test is to evaluate your technical abilities in a few aspects of software engineering. It allows common grounds for discussion in the interview that takes place after you complete the test. The test is a home test and you are encouraged to use any resources at your disposal - online or consulting friends, etc. Please submit your code via a Github repository and send us a link to the Github repo. You should be able to answer these 3 questions within 3 hours.

1. JSON is a document format used to encode information that is both human-readable and machine-readable. JSON format is explained at http://json.org. Please write a JSON parser that accepts an input JSON string and produces a Map output structure. Do not use any existing library to do the parsing.

   Example input JSON:

   ```
   {
       "debug" : "on".
       "window" : {
          "title" : "sample",
          "size": 500
       }
   }
   ```

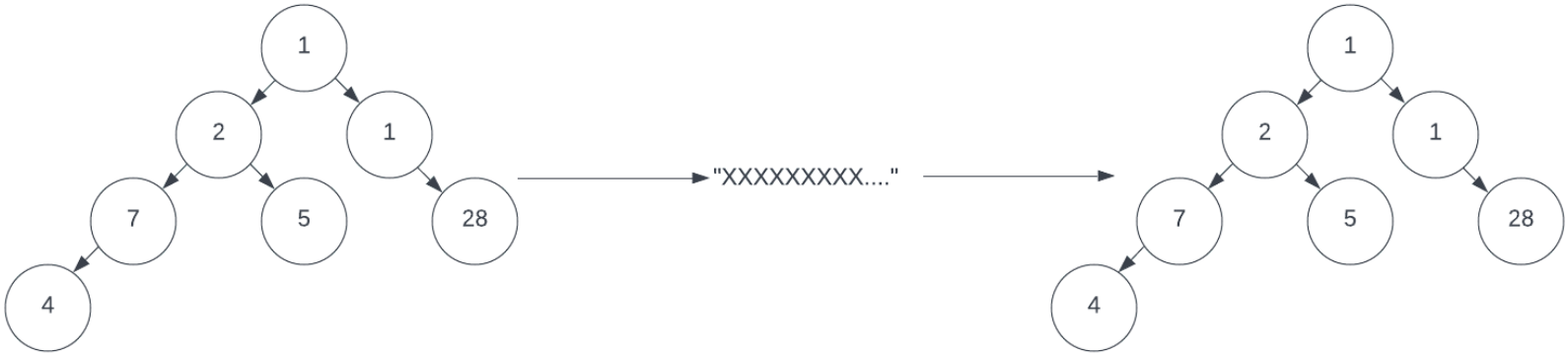
   The parser may be a static function:

   ```
   public class JSONParser {
       public static Map<String,Object> parse(String json) {
       ...
       }
   }
   ```

   The corresponding output for the input JSON should be:

   ```
   Map<string, Object> output = JsonParser.parse(input);
   assert output.get("debug").equals("on");
   assert (Map<string, Object>(output.get("window")).get("title").equals("sample")
   assert (Map<String, Object>(output.get("window")).get("size").equals(500)
   ```

2. Assuming you have a binary tree of integers, come up with an algorithm to serialize and deserialize it.
   Assumptions:
   a. Serialization should be to a String and from a String.
   b. There are no cyclic connections in the tree



Model:

```
public class Node {
    Node left;
    Node right;
    int num;
}

public interface TreeSerializer {
    String serialize(Node root);

    Node deserialize(String str);
}
```

I. Implement a `TreeSerializer` given the above assumptions.
II. Implement a `TreeSerializer` that takes into account cyclic trees.
III. Suggest changes that should be done in order to support any data type (as opposed to only an `int` data type)

3. Please design three MySQL tables to store information about:
    a. product: the “product” tables should include the name, category of the product; it should also provide information to tell us when this product was added to the table, and who added this product.
    b. product price: this table will store the current price information of each product and we should be able to join it with the product table. It should include the price, current discount percent(default to 0), the updated time and who updated it.
    c. product price change log: this table will store the old and new value of the “product price” table that’s impacted by any insert/update/delete and contains information of who and when the operation was performed.

    Please list the table creation scripts in proper MySQL syntax for these three tables and a query to join “product” table and “product price” table together to show the product name, category, price, and who/when it gets updated.