Description | Solution | Discuss (311) | Submissions

Java | Autocomplete

271. Encode and Decode Strings

Medium 719 200 Add to List Share

Design an algorithm to encode **a list of strings** to **a string**. The encoded string is then sent over the network and is decoded back to the original list of strings.

Machine 1 (sender) has the function:

```
string encode(vector<string> strs) {
  // ... your code
  return encoded_string;
}
```

Machine 2 (receiver) has the function:

```
vector<string> decode(string s) {
  //... your code
  return strs;
}
```

So Machine 1 does:

```
string encoded_string = encode(strs);
```

and Machine 2 does:

```
vector<string> strs2 = decode(encoded_string);
```

`strs2` in Machine 2 should be the same as `strs` in Machine 1.

Implement the `encode` and `decode` methods.

You are not allowed to solve the problem using any serialize methods (such as `eval`).

Example 1:

```
Input: dummy_input = ["Hello","World"]
Output: ["Hello","World"]
Explanation:
Machine 1:
Codec encoder = new Codec();
String msg = encoder.encode(strs);
Machine 1 ---msg---> Machine 2

Machine 2:
Codec decoder = new Codec();
String[] strs = decoder.decode(msg);
```

Example 2:

```
Input: dummy_input = [""]
Output: [""]
```

Constraints:

- `1 <= strs.length <= 200`
- `0 <= strs[i].length <= 200`
- `strs[i]` contains any possible characters out of `256` valid ASCII characters.

Follow up: Could you write a generalized algorithm to work on any possible set of characters?

Accepted 81,508 | Submissions 228,704

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Facebook | 4

Related Topics

Array String Design

Similar Questions

Count and Say Medium

Serialize and Deserialize Binary Tree Hard

String Compression Medium

Count Binary Substrings Easy

```java
public class Codec {
  // Encodes string length to bytes string
  public String intToString(String s) {
    int x = s.length();
    char[] bytes = new char[4];
    for(int i = 3; i > -1; --i) {
      bytes[3 - i] = (char) (x >> (i * 8) & 0xff);
    }
    return new String(bytes);
  }

  // Encodes a list of strings to a single string.
  public String encode(List<String> strs) {
    StringBuilder sb = new StringBuilder();
    for(String s: strs) {
      sb.append(intToString(s));
      sb.append(s);
    }
    return sb.toString();
  }

  // Decodes bytes string to integer
  public int stringToInt(String bytesStr) {
    int result = 0;
    for(char b : bytesStr.toCharArray())
      result = (result << 8) + (int)b;
    return result;
  }

  // Decodes a single string to a list of strings.
  public List<String> decode(String s) {
    int i = 0, n = s.length();
    List<String> output = new ArrayList();
    while (i < n) {
      int length = stringToInt(s.substring(i, i + 4));
      i += 4;
      output.add(s.substring(i, i + length));
      i += length;
    }
    return output;
  }
}
```