Problem 535: Encode and Decode TinyURL

Problem Information

Difficulty: Medium
Acceptance Rate: 86.50%
Paid Only: No
Tags: Hash Table, String, Design, Hash Function

Problem Description

> Note: This is a companion problem to the [System > Design](https://leetcode.com/discuss/interview-question/system-design/) > problem: [Design TinyURL](https://leetcode.com/discuss/interview- > question/124658/Design-a-URL-Shortener-\(-TinyURL-\)-System/).

TinyURL is a URL shortening service where you enter a URL such as `https://leetcode.com/problems/design-tinyurl` and it returns a short URL such as `http://tinyurl.com/4e9iAk`. Design a class to encode a URL and decode a tiny URL.

There is no restriction on how your encode/decode algorithm should work. You just need to ensure that a URL can be encoded to a tiny URL and the tiny URL can be decoded to the original URL.

Implement the `Solution` class:

* `Solution()` Initializes the object of the system. * `String encode(String longUrl)` Returns a tiny URL for the given `longUrl`. * `String decode(String shortUrl)` Returns the original long URL for the given `shortUrl`. It is guaranteed that the given `shortUrl` was encoded by the same object.

Example 1:

Input: url = "https://leetcode.com/problems/design-tinyurl" **Output:** "https://leetcode.com/problems/design-tinyurl" **Explanation:** Solution obj = new Solution(); string tiny = obj.encode(url); // returns the encoded tiny url. string ans = obj.decode(tiny); // returns the original url after decoding it.

Constraints:

* `1 <= url.length <= 104` * `url` is guranteed to be a valid URL.

Code Snippets

C++:

```
class Solution {
public:

// Encodes a URL to a shortened URL.
string encode(string longUrl) {

}

// Decodes a shortened URL to its original URL.
string decode(string shortUrl) {

}
};

// Your Solution object will be instantiated and called as such:
// Solution solution;
// solution.decode(solution.encode(url));
```

Java:

```
public class Codec {

// Encodes a URL to a shortened URL.
public String encode(String longUrl) {

}

// Decodes a shortened URL to its original URL.
public String decode(String shortUrl) {

}
}
```

```
// Your Codec object will be instantiated and called as such:
// Codec codec = new Codec();
// codec.decode(codec.encode(url));
```

Python3:

```
class Codec:

def encode(self, longUrl: str) -> str:
"""Encodes a URL to a shortened URL.
"""


def decode(self, shortUrl: str) -> str:
"""Decodes a shortened URL to its original URL.
"""


# Your Codec object will be instantiated and called as such:
# codec = Codec()
# codec.decode(codec.encode(url))
```