lee215 ★ 47713 Last Edit: November 16, 2019 10:00 PM 1.4K VIEWS

50

Solution 1: Recursion idea

The following sequence can be built up form the ealier result.
So I search index of the prefix part
For example:
`f(5) = "10"`
`f(6) = "11"`
The prefix are both `f(2) = "1"`

so we found that `f(n)` has `f((n - 1) / 2)` as prefix.

Java:

```
public String encode(int n) {
    return n > 0 ? encode((n - 1) / 2) + "10".charAt(n % 2) : "";
}
```

C++:

```
string encode(int n) {
    return n > 0 ? encode((n - 1) / 2) + "10"[n % 2] : "";
}
```

Python:

```
def encode(self, n):
    return self.encode((n - 1) / 2) + '10'[n % 2] if n else ""
```

Solution 2: Binary of n + 1

Assume `g(n) = "1" + f(n)`
we can find:
`g(0) = "1" g(1) = "10" g(2) = "111" g(3) = "100" g(4) = "101" g(5) = "110" g(6) = "111"`

Now everything is obvious:
`g(n) = binary(n + 1)`
`"1" + f(n) = binary(n + 1)`
`f(n) = binary(n + 1).substring(1)`

Java:

```
public String encode(int n) {
    return Integer.toBinaryString(n + 1).substring(1);
}
```

Python:

```
def encode(self, n):
    return bin(n + 1)[3:]
```

Complexity

Time `O(logN)`
Space `O(logN)`

More

Thanks for upvotes and followers.
weibo leecode