Menu

by Micro Admin

> *I'm working through algoexpert.io coding challenges and I'm having trouble undersatnding the suggested solution to one of the questions titled* ***Non-Constructible Change***
>
> *Here's the challenge question:*

*Given an array of positive integers representing the values of coins in your possession, write a function that returns the minimum amount of change (the minimum sum of money) that you **cannot** create. The given coins can have any positive integer value and aren't necessarily unique (i.e., you can have multiple coins of the same value).*

For example, if you're given coins = [1, 2, 5], the minimum amount of change that you can't create is 4. If you're given no coins, the minimum amount of change that you can't create is 1.

```
  coins = coins.sort((a, b) => a - b); // O(nlogn) time operation
  let change = 0;

  for (coin of coins) {
    if (coin > change + 1) return change + 1;
    change += coin;
  }

  return change + 1;
}
```

My problem

I am not completely sure how did the author of the solution come up with the intuition that

```
if the current coin is greater than `change + 1`, the smallest impossible change is equal to `change + 1
```

I can see how it tracks, and indeed the algorithm passes all tests, but I'd like to know more about a process I could use to devise this rule.

Thank you for taking the time to read the question!

Answer

This one took me a while too, but this was how I made sense of it:

Assume you've proven you can make 1-8 cents.

You go to the next iteration and want to know if you can make 9 cents. So you iterate to the next new coin in the sorted list.

if newCoin < 9:

- You know for a fact that you can make 9 cents.
- Example if the new coin is 5: Use that new coin and subtract it from the total you're trying to make. 9 – 5 = 4. Then however way you made 4 previously just do that again. (You've already proven you can make 1-8 cents)

if newCoin == 9:

if newCoin > 9:

- You know for a fact that you CANNOT make 9 cents
- This is because **you can't use the new coin**. For example, a 10 cent coin is useless to you when you're trying to make 9 cents since it's too big (can't make 9)
- **You're also screwed if you don't use the new coin** because if you add up all the coins you've seen so far, you've only been able to make 8 (can't make 9)

And that's where the change + 1 comes from. (your variable change = 8)

```
if the current coin is greater than `change + 1`, the smallest impossible change is equal to `change + 1`.
```

Attribution

Source : Link , Question Author : Daniel Kaczmarczyk , Answer Author : Jack Windensky

Uncategorized

‹ Enforce immutable collections in a Java record?

› RuntimeError: cuDNN error: CUDNN_STATUS_NOT_INITIALIZED using pytorch

Leave a Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search

Search

Latest Post

Testing regex with Java

Converting image to binary array (blob) with HTML5

How to disable event notifications for group I'm not in?

Minecraft

How can I test for a player at a given x and z but at any height?

I Keep Spawning In The Middle Of Nowhere On A Map

Diablo

How do I effectively use multiple skills?

What is the best method for farming for a Stone of Jordan? [duplicate]

Are you rewarded for playing Diablo II in hardcore mode?