Problem List

Run Submit

Description | Editorial | Solutions | Submissions

800. Similar RGB Color Premium

Solved

Easy Topics Companies

The red-green-blue color `"#AABBCC"` can be written as `"#ABC"` in shorthand.

- For example, `"#15c"` is shorthand for the color `"#1155cc"`.

The similarity between the two colors `"#ABCDEF"` and `"#UVWXYZ"` is $-(AB - UV)^2 - (CD - WX)^2 - (EF - YZ)^2$.

Given a string `color` that follows the format `"#ABCDEF"`, return a string represents the color that is most similar to the given color and has a shorthand (i.e., it can be represented as some `"#XYZ"`).

Any answer which has the same highest similarity as the best answer will be accepted.

Example 1:

> **Input:** color = "#09f166"
> **Output:** "#11ee66"
> **Explanation:**
> The similarity is $-(0x09 - 0x11)^2 -(0xf1 - 0xee)^2 - (0x66 - 0x66)^2 = -64 -9 -0 = -73$.
> This is the highest among any shorthand color.

Example 2:

> **Input:** color = "#4e3fe1"
> **Output:** "#5544dd"

Constraints:

- `color.length == 7`
- `color[0] == '#'`
- `color[i]` is either digit or character in the range `['a', 'f']` for `i > 0`.

Seen this question in a real interview before? 1/5

Yes No

Accepted 17.2K | Submissions 25.6K | Acceptance Rate 67.4%

Topics

Companies

Discussion (1)

108 1

Code

Python3 • Auto

```python
class Solution:
    def similarRGB(self, color: str) -> str:

        # We first split color into an array containing its three two-letter parts
        # and convert each value to an int of base 16 (hexadecimal)

        color = [int(color[i:i+2], 16) for i in range(1, len(color), 2)]


        # How do we find the closest number which can be written in shorthand?
        # Well, let's look at the pattern: 0x11, 0x22, ..., 0xFF
        # We can see that all of these numbers are divisible by 0x11 (17 in decimal)

        # To minimize this difference, we divide the value of each color by 0x11, round it, and convert to hex
        # This produces a shorthand string like 0x1, 0x2, ..., 0xF
        # But we want to return the full string, so we take the 3rd char (1,2,...,F) and double it (11,22,...FF)

        # We concatenate these strings and add '#' to the front to make the full 7-character color, and then return this string

        return '#'+''.join([hex(round(c/int('0x11', 16)))[2]*2 for c in color])
```

Saved Ln 14, Col 35

Testcase | Test Result

Accepted Runtime: 32 ms

• Case 1 • Case 2

Input

color =
"#09f166"

Output

"#11ee66"

Expected

"#11ee66"