

lee215 ★ 47725 Last Edit: May 7, 2019 7:55 AM 1.4K VIEWS

21 Assume $n \% k = r$,
we will fill all positions of i with $i \% k = r$,
then the positions with $i \% k = r - 1$,

```
def rearrangeString(self, s, k):
    n = len(s)
    if not k: return s
    count = collections.Counter(s)
    maxf = max(count.values())
    if (maxf - 1) * k + count.values().count(maxf) > len(s):
        return ""
    res = list(s)
    i = (n - 1) % k
    for c in sorted(count, key=lambda i: -count[i]):
        for j in range(count[c]):
            res[i] = c
            i += k
            if i >= n:
                i = (i - 1) % k
    return "".join(res)
```

Comments: 8

Best Most Votes Newest to Oldest Oldest to Newest

Type comment here... (Markdown is supported)

Post

madno ★ 246 Last Edit: December 14, 2019 2:28 PM

Nice.

Let's see some examples.

One might start out with the naive approach, like this:

input: "abcd abcd abde ac" (added some spaces just like at iban bank account numbers, for easier readability)
k: 4

We have $k=4$ buckets. Let's assign the characters naively from left to right to each bucket in decreasing frequency order, and when running out of buckets, we can restart from the first bucket but next unoccupied seat:

a--- a--- a--- a-

ab-- ab-- ab-- a- overflow b restarted from first bucket next seat

[Read More](#)

4 Reply

otoc ★ 961 Last Edit: December 15, 2019 12:42 AM

Thanks for sharing this concise and beautiful solution. I have similar ideas when I tried to solve this problem. Please see my post [Different Python Greedy solutions](#).

1 Reply

stockfish ★ 169 June 28, 2019 3:28 AM

求解释

0 Show 1 reply Reply