Problem #1309: Decrypt String from Alphabet to Integer Mapping

<https://leetcode.com/problems/decrypt-string-from-alphabet-to-integer-mapping/description/>

**My Solution:**

1. Initialize res as an empty list to hold the result. Note: Don’t use a string because strings are immutable.
2. Initialize index I to 0.
3. Iterate through the length of the string s.
4. If i + 2 is less then length of the string and the character at index I +2 is “#:, then the decoded letter is ASCII character of the slice of string from index I to I + 2 shifted by the value of ordinal of letter ‘a’ – 1 since ‘a’ is 1 (not 0). Increment the index by 3.
5. Otherwise, the decoded letter is the decoded letter is ASCII character of the character at index I in s shifted by the value of ordinal of letter ‘a’ – 1 since ‘a’ is 1 (not 0). Increment the index by 1.
6. Append the letter to res.
7. Return the join of res with no space to form a string.

class Solution:

def freqAlphabets(self, s: str) -> str:

res = []

i = 0

while i < len(s):

if i + 2 < len(s) and s[i + 2] == '#':

letter = chr(ord('a') - 1 + int(s[i : i + 2])) #since a = 1

i += 3

else:

letter = chr(ord('a') - 1 + int(s[i]))

i += 1

res.append(letter)

return "".join(res)