

Live Programming for Validation

```

3 def dominant_bigram(s):
4     """
5     Return the most common bigram in string s.
6     """
7     ## ---
8     res = ''
9     bigrams = {}
10    for i in range(0, len(s) - 1):
11        bigram = s[i] + s[i + 1]
12        if bigram not in bigrams:
13            bigrams[bigram] = 0
14        bigrams[bigram] += 1
15    max_value = max(bigrams.values())
16    for bigram in bigrams:
17        if bigrams[bigram] == max_value:
18            res = bigram
19    ## ---
20
21    return res
22
23
24 dominant_bigram("agctagta")

```

```

# bigram
0 'ag'
1 'gc'
2 'ct'
3 'ta'
4 'ag'
5 'gt'
6 'ta'

```

```

# bigrams
0 {'ag': 0}
1 {'ag': 1, 'gc': 0}
2 {'ag': 1, 'gc': 1, 'ct': 0}
3 {'ag': 1, 'gc': 1, 'ct': 1, 'ta': 0}
4
5 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 1, 'gt': 0}
6

```

```

# bigrams
0 {'ag': 1}
1 {'ag': 1, 'gc': 1}
2 {'ag': 1, 'gc': 1, 'ct': 1}
3 {'ag': 1, 'gc': 1, 'ct': 1, 'ta': 1}
4 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 1}
5 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 1, 'gt': 1}
6 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 2, 'gt': 1}

```

```

# max_value
2

```

```

# res
0 'ag'
1
2
3 'ta'
4

```



```

2 def dominant_bigram(s):
3     '''
4     Return the most common bigram in string s.
5     '''
6     ## ---
7     res = ''
8     bigrams = {}
9     for i in range(0, len(s) - 1):
10         bigram = s[i] + s[i + 1]
11         if bigram not in bigrams:
12             bigrams[bigram] = 0
13         bigrams[bigram] += 1
14     max_value = max(bigrams.values())
15     for bigram in bigrams:
16         if bigrams[bigram] == max_value:
17             res = bigram
18     ## ---
19
20     return res

```

```

24 dominant_bigram("agctagta")

```

```

• # bigram
0 'ag'
1 'gc'
2 'ct'
3 'ta'
4 'ag'
5 'gt'
6 'ta'

```

```

• # bigrams
0 {'ag': 0}
1 {'ag': 1, 'gc': 0}
2 {'ag': 1, 'gc': 1, 'ct': 0}
3 {'ag': 1, 'gc': 1, 'ct': 1, 'ta': 0}
4
5 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 1, 'gt': 0}
6

```

```

• # bigrams
0 {'ag': 1}
1 {'ag': 1, 'gc': 1}
2 {'ag': 1, 'gc': 1, 'ct': 1}
3 {'ag': 1, 'gc': 1, 'ct': 1, 'ta': 1}
4 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 1}
5 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 1, 'gt': 1}
6 {'ag': 2, 'gc': 1, 'ct': 1, 'ta': 2, 'gt': 1}

```

```

• max_value
2

```

```

• # res
0 'ag'
1
2
3 'ta'
4

```

Suggestion 2

Preview

```

res = {}
s = s.lower()
for i in range(0, len(s)):
    if s[i] in res:
        res[s[i]] += 1
    else:
        res[s[i]] = 1
print(res)
for i in res.keys():
    if res[i] < 2:
        res.pop(i)
print(res)

```

Suggestion 3

Preview

```

res = ''
bigrams = {}
for i in range(0, len(s) - 1):

```

