


```

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")

```

Validation of AI-Generated Content is Burden

Validation by exclusion is great but...

- manual

time-consuming

-filed-with-context-switching



AI-Generated



Validation of AI-Generated Code is Burdensome

```
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")
```

AI-Generated

Validation by execution is great but...

- manual
- time-consuming
- filled with context-switching

What might help validate AI-generated code?