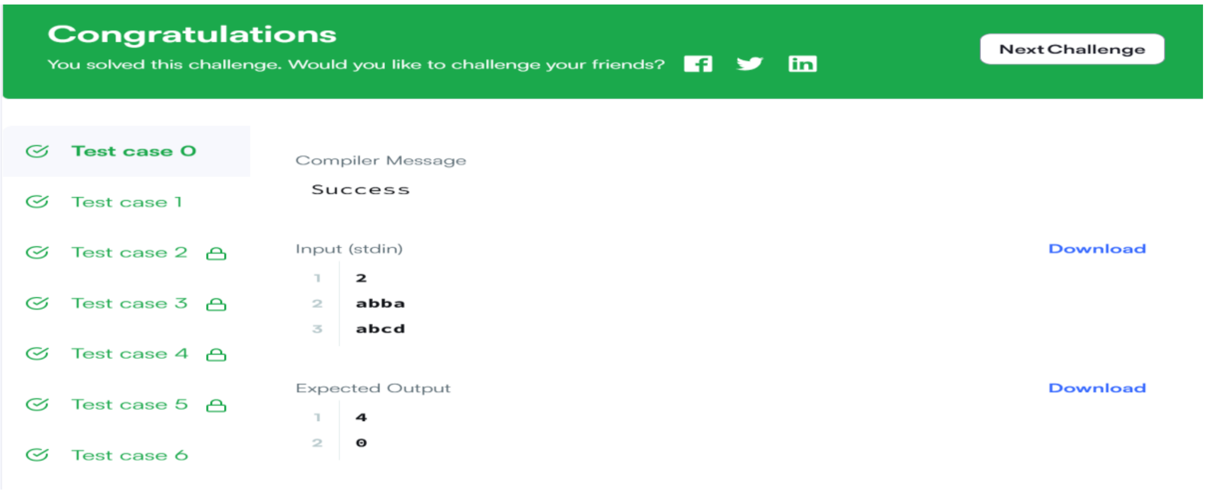
## DAY 46

## Hacker Rank

```
#!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'sherlockAndAnagrams' function below.
#
# The function is expected to return an INTEGER.
# The function accepts STRING s as parameter.
#

def sherlockAndAnagrams(s):
    from collections import defaultdict

    substr_count = defaultdict(int)

    # Generate all substrings
    for start in range(len(s)):
        freq = [0] * 26  # frequency map for a-z
        for end in range(start, len(s)):
            # Update the frequency map
            char_index = ord(s[end]) - ord('a')
            freq[char_index] += 1
            # Use tuple of frequency counts as a key
            substr_count[tuple(freq)] += 1

    # Count anagrammatic pairs
    total_pairs = 0
    for count in substr_count.values():
        if count > 1:
            total_pairs += (count * (count - 1)) // 2

    return total_pairs

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    q = int(input().strip())

    for q_itr in range(q):
        s = input()

        result = sherlockAndAnagrams(s)

        fptr.write(str(result) + '\n')

    fptr.close()
```



**Congratulations**

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

Input (stdin)

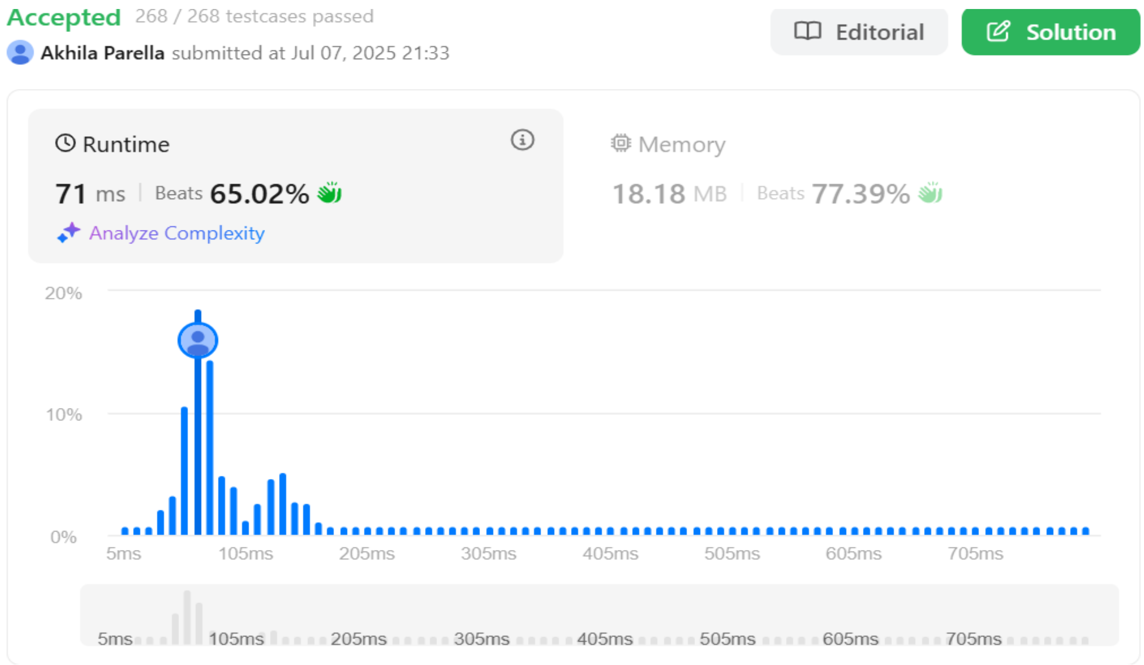
```
2
abba
abcd
```

Expected Output

```
4
0
```

## Leetcode

```
class Solution:
    def minWindow(self, s: str, t: str) -> str:
        need, missing = Counter(t), len(t)
        left = start = end = 0
        for right, c in enumerate(s, 1):
            missing -= need[c] > 0
            need[c] -= 1
            if missing == 0:
                while left < right and need[s[left]] < 0:
                    need[s[left]] += 1
                    left += 1
                if not end or right - left < end - start:
                    start, end = left, right
                need[s[left]] += 1
                missing += 1
                left += 1
        return s[start:end]
```



Accepted 268 / 268 testcases passed

Akhila Parella submitted at Jul 07, 2025 21:33

Runtime

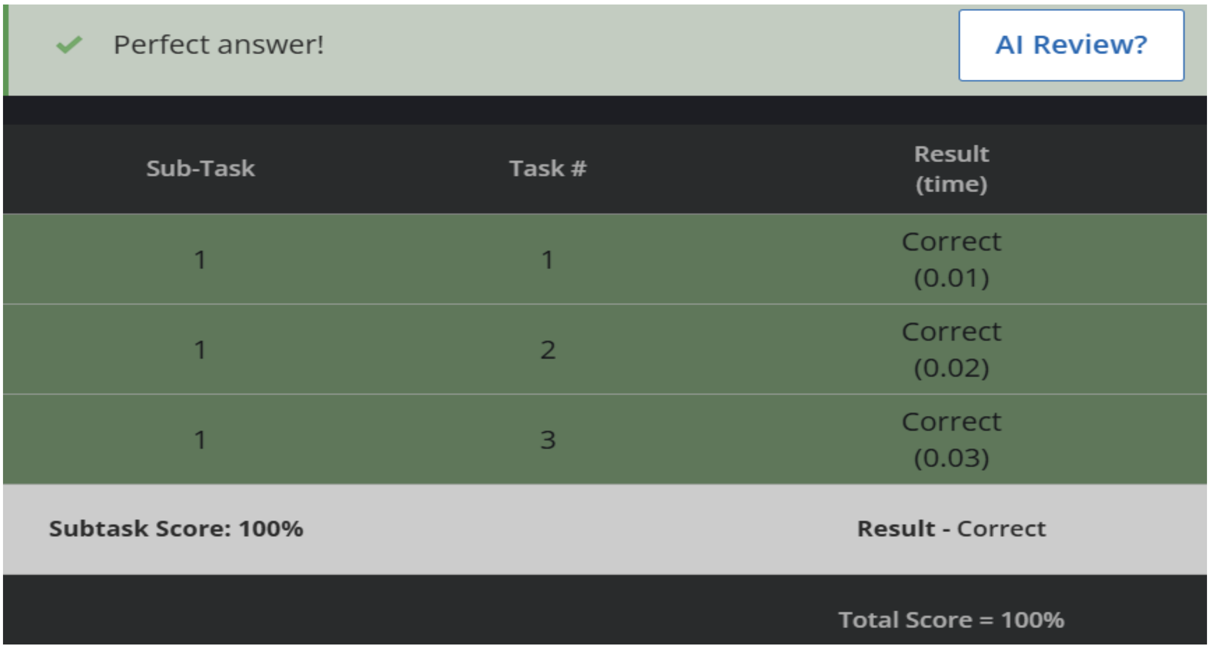
71 ms | Beats 65.02%

Memory

18.18 MB | Beats 77.39%

## Code Chef

```
t = int(input())


while t > 0:
    s = input()
    vowels = "aeiou"
    count = 0
    happy = False


    for c in s:
        if c in vowels:
            count += 1
            if count > 2:
                happy = True
                break
        else:
            count = 0


    print("HAPPY" if happy else "SAD")
    t -= 1
```



✓ Perfect answer!

| Sub-Task | Task # | Result<br>(time) |
|---|---|---|
| 1 | 1 | Correct<br>(0.01) |
| 1 | 2 | Correct<br>(0.02) |
| 1 | 3 | Correct<br>(0.03) |

Subtask Score: 100% Result - Correct

Total Score = 100%