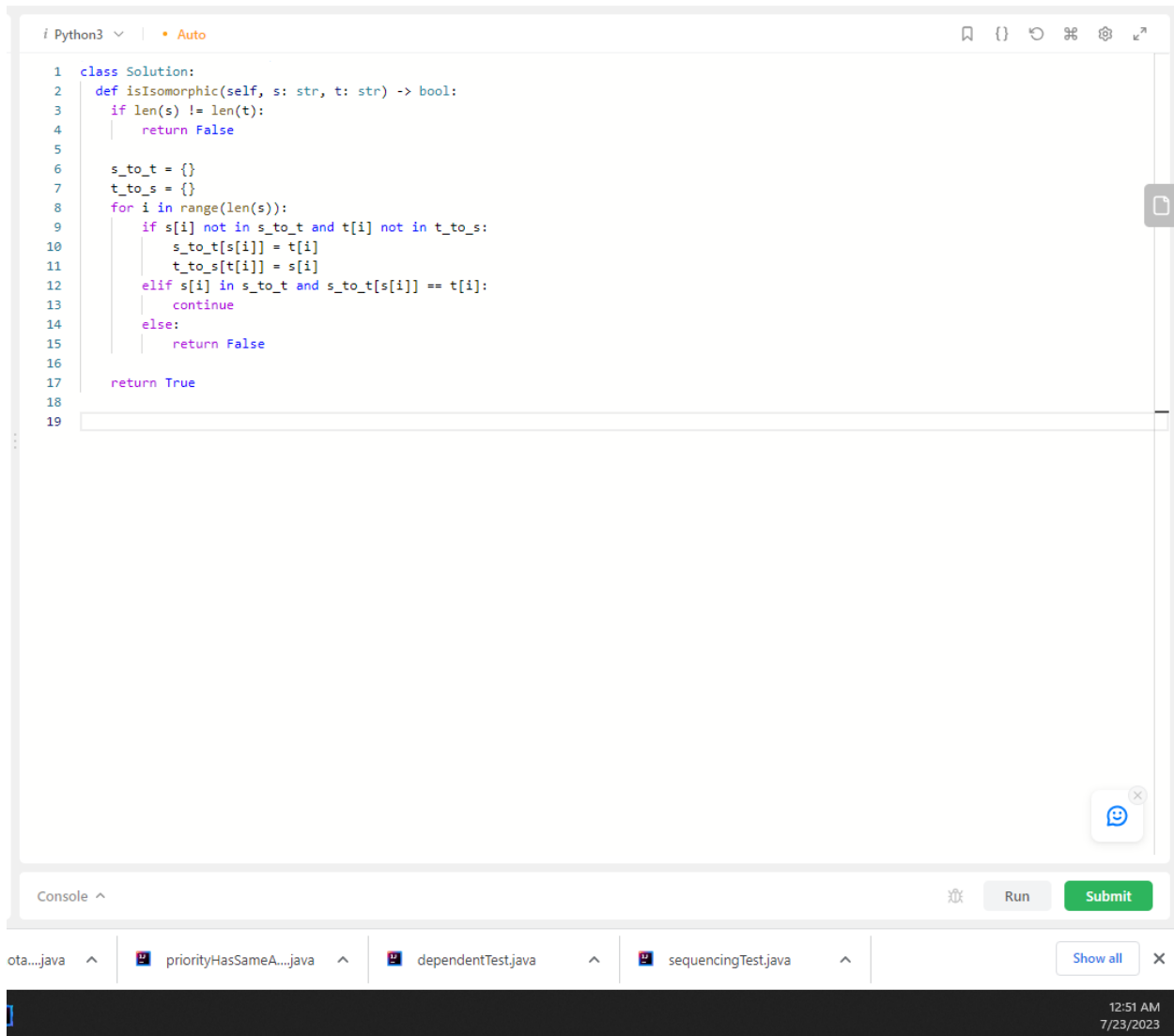


Step 1: Study 205. Isomorphic Strings - (local copy)



```
1 class Solution:
2     def isIsomorphic(self, s: str, t: str) -> bool:
3         if len(s) != len(t):
4             return False
5
6         s_to_t = {}
7         t_to_s = {}
8         for i in range(len(s)):
9             if s[i] not in s_to_t and t[i] not in t_to_s:
10                 s_to_t[s[i]] = t[i]
11                 t_to_s[t[i]] = s[i]
12             elif s[i] in s_to_t and s_to_t[s[i]] == t[i]:
13                 continue
14             else:
15                 return False
16
17         return True
18
19
```

ota....java ^ | priorityHasSameA....java ^ | dependentTest.java ^ | sequencingTest.java ^ | Show all x

12:51 AM
7/23/2023

Step 2: Following the examples shown on Hash Table to manually solve the LeeCode problem with test data 'Input: s = "egg", t = "add"; Output: true'

This step is to demonstrate that you know the concept of the algorithm by showing something manually created such as diagrams.

1) create a hash table between s and t; create a hash table between t and s.

s_to_t = {e: a, g: d}

t_to_s = {a: e, d: g}

2) since s_to_t[key] = t_to_s[value] for all items at the same position, they are isomorphic.

Step 3: Ask ChatGPT : "How to use Hash Table approach to implement a Python code to solve the LeetCode question: '205. Isomorphic Strings', with the test data: 'Input: s = "egg", t = "add"; Output: true'"

To solve the LeetCode question "205. Isomorphic Strings" using a Hash Table approach in Python, we need to check whether two strings are isomorphic, which means that the characters in one string can be replaced to get the other string. For example, in the given test data "egg" and "add," we can replace 'e' with 'a' and 'g' with 'd' to transform "egg" into "add."

Here's the Python code using the Hash Table approach to solve the problem:

```
python 📄 Copy code  
  
def is_isomorphic(s, t):  
    if len(s) != len(t):  
        return False  
  
    char_map_s = {} # To store the mapping of characters from s to t  
    char_map_t = {} # To store the mapping of characters from t to s  
  
    for i in range(len(s)):  
        char_s, char_t = s[i], t[i]
```

```

    # Check if the characters are already mapped
    if char_s in char_map_s and char_map_s[char_s] != char_t:
        return False

    if char_t in char_map_t and char_map_t[char_t] != char_s:
        return False

    # Create the mapping if it does not exist
    char_map_s[char_s] = char_t
    char_map_t[char_t] = char_s

    return True

# Test the function with the given input data
s = "egg"
t = "add"
output = is_isomorphic(s, t)
print(output) # Output: True

```

The function `is_isomorphic()` takes two strings, `s` and `t`, as input and returns `True` if they are isomorphic, otherwise returns `False`. It uses two hash tables (`char_map_s` and `char_map_t`) to keep track of the character mappings between the two strings.

In the given test data, "egg" and "add" are isomorphic because they can be mapped as follows: 'e' -> 'a' and 'g' -> 'd'. Therefore, the output is `True`.

Step 4: Implement the Python code created by ChatGPT

The image shows a code editor window with a file named `test.py`. The code defines a function `is_isomorphic(s, t)` that checks if two strings are isomorphic. It uses two dictionaries, `char_map_s` and `char_map_t`, to store character mappings. The function returns `True` if the strings are isomorphic and `False` otherwise. Below the function, there is a test case with `s = "egg"` and `t = "add"`, which results in `True`.

```
test.py > ...
1  def is_isomorphic(s, t):
2      if len(s) != len(t):
3          return False
4
5      char_map_s = {} # To store the mapping of characters from s to t
6      char_map_t = {} # To store the mapping of characters from t to s
7
8      for i in range(len(s)):
9          char_s, char_t = s[i], t[i]
10
11         # Check if the characters are already mapped
12         if char_s in char_map_s and char_map_s[char_s] != char_t:
13             return False
14
15         if char_t in char_map_t and char_map_t[char_t] != char_s:
16             return False
17
18         # Create the mapping if it does not exist
19         char_map_s[char_s] = char_t
20         char_map_t[char_t] = char_s
21
22     return True
23
24     # Test the function with the given input data
25     s = "egg"
26     t = "add"
27     output = is_isomorphic(s, t)
28     print(output) # Output: True
29
```

The output panel shows the command `python -u "d:\MSCS\CS455 Algorithm\w10\test.py"` being executed, resulting in `True`. The status bar at the bottom indicates the file is at line 29, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and is being edited with Python 3.11.3 64-bit.

Step 5: Test the Python code with all the test cases provided by 205. Isomorphic Strings - (local copy)

test.py ×

test.py > ...

```
1 def is_isomorphic(s, t):
2     if len(s) != len(t):
3         return False
4
5     char_map_s = {} # To store the mapping of characters from s to t
6     char_map_t = {} # To store the mapping of characters from t to s
7
8     for i in range(len(s)):
9         char_s, char_t = s[i], t[i]
10
11         # Check if the characters are already mapped
12         if char_s in char_map_s and char_map_s[char_s] != char_t:
13             return False
14
15         if char_t in char_map_t and char_map_t[char_t] != char_s:
16             return False
17
18         # Create the mapping if it does not exist
19         char_map_s[char_s] = char_t
20         char_map_t[char_t] = char_s
21
22     return True
23
24 # Test the function with the given input data
25 s = "egg"
26 t = "add"
27 output = is_isomorphic(s, t)
28 print(output) # Output: True
29
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code

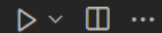
```
[Running] python -u "d:\MSCS\CS455 Algorithm\w10\test.py"
True

[Done] exited with code=0 in 0.146 seconds
```

Column Selection Ln 29, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.11.3 64-bit Go Live

8:02 PM 7/23/2023

test.py ×



test.py > ...

```
1 def is_isomorphic(s, t):
2     if len(s) != len(t):
3         return False
4
5     char_map_s = {} # To store the mapping of characters from s to t
6     char_map_t = {} # To store the mapping of characters from t to s
7
8     for i in range(len(s)):
9         char_s, char_t = s[i], t[i]
10
11         # Check if the characters are already mapped
12         if char_s in char_map_s and char_map_s[char_s] != char_t:
13             return False
14
15         if char_t in char_map_t and char_map_t[char_t] != char_s:
16             return False
17
18         # Create the mapping if it does not exist
19         char_map_s[char_s] = char_t
20         char_map_t[char_t] = char_s
21
22     return True
23
24 # Test the function with the given input data
25 s = "foo"
26 t = "bar"
27 output = is_isomorphic(s, t)
28 print(output) # Output: False
29
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Code



[Running] python -u "d:\MSCS\CS455 Algorithm\w10\test.py"

False

[Done] exited with code=0 in 0.133 seconds

Column Selection

Ln 28, Col 31

Spaces: 4

UTF-8

CRLF

{ } Python

3.11.3 64-bit



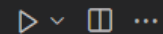
Go Live



8:04 PM

7/23/2023

test.py X



test.py > ...

```
1 def is_isomorphic(s, t):
2     if len(s) != len(t):
3         return False
4
5     char_map_s = {} # To store the mapping of characters from s to t
6     char_map_t = {} # To store the mapping of characters from t to s
7
8     for i in range(len(s)):
9         char_s, char_t = s[i], t[i]
10
11         # Check if the characters are already mapped
12         if char_s in char_map_s and char_map_s[char_s] != char_t:
13             return False
14
15         if char_t in char_map_t and char_map_t[char_t] != char_s:
16             return False
17
18         # Create the mapping if it does not exist
19         char_map_s[char_s] = char_t
20         char_map_t[char_t] = char_s
21
22     return True
23
24 # Test the function with the given input data
25 s = "paper"
26 t = "title"
27 output = is_isomorphic(s, t)
28 print(output) # Output: True
29
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Code



[Running] python -u "d:\MSCS\CS455 Algorithm\w10\test.py"

True

[Done] exited with code=0 in 0.135 seconds

Column Selection

Ln 27, Col 12

Spaces: 4

UTF-8

CRLF

{ } Python

3.11.3 64-bit

Go Live



8:04 PM

7/23/2023