1. Python | Convert a list of Tuples into Dictionary
2. Python counter and dictionary intersection example (Make a string using deletion and rearrangement)
3. Python dictionary, set and counter to check if frequencies can become same
4. Scraping And Finding Ordered Words In A Dictionary using Python
5. Possible Words using given characters in Python
6. Python – Keys associated with Values in Dictionary
7. Python program to Find the size of a Tuple
8. Python – Maximum and Minimum K elements in Tuple
9. Create a list of tuples from given list having number and its cube in each tuple
10. Python – Adding Tuple to List and vice – versa
11. Python – Closest Pair to Kth index element in Tuple

Solve

Sure, I can provide solutions for each of these problems. I'll go through them one by one:

### 1. Convert a list of Tuples into Dictionary

You can convert a list of tuples into a dictionary using a simple comprehension or the `dict()` constructor.

```python

# List of tuples

list\_of\_tuples = [('a', 1), ('b', 2), ('c', 3)]

# Converting to dictionary

dictionary = dict(list\_of\_tuples)

print(dictionary)

```

### 2. Python Counter and Dictionary Intersection Example (Make a string using deletion and rearrangement)

To solve this problem, you can use the `collections.Counter` to find the intersection between two dictionaries.

```python

from collections import Counter

def can\_form\_string(s1, s2):

counter1 = Counter(s1)

counter2 = Counter(s2)

intersection = counter1 & counter2

result = ''.join(intersection.elements())

return result

# Example usage

s1 = "abccc"

s2 = "cbad"

print(can\_form\_string(s1, s2))

```

### 3. Python Dictionary, Set, and Counter to Check if Frequencies Can Become Same

This problem can be approached by using a `Counter` to check the frequency of each character and then analyzing if you can rearrange the frequencies to make them the same.

```python

from collections import Counter

def can\_make\_frequencies\_same(s):

freq = Counter(s)

freq\_counts = Counter(freq.values())

if len(freq\_counts) == 1:

return True

elif len(freq\_counts) == 2:

key1, key2 = freq\_counts.keys()

if (key1 == 1 and freq\_counts[key1] == 1) or (key2 == 1 and freq\_counts[key2] == 1):

return True

if abs(key1 - key2) == 1 and (freq\_counts[key1] == 1 or freq\_counts[key2] == 1):

return True

return False

# Example usage

s = "aabbcc"

print(can\_make\_frequencies\_same(s))

```

### 4. Scraping And Finding Ordered Words In A Dictionary using Python

You can use the `requests` and `BeautifulSoup` libraries to scrape a website, and then check if the words are ordered.

```python

import requests

from bs4 import BeautifulSoup

def scrape\_and\_find\_ordered\_words(url):

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

words = soup.get\_text().split()

ordered\_words = [word for word in words if word == ''.join(sorted(word))]

return ordered\_words

# Example usage

url = "https://example.com"

print(scrape\_and\_find\_ordered\_words(url))

```

### 5. Possible Words using Given Characters in Python

You can use the `itertools.permutations` or simply a recursive approach to find all possible words that can be formed from given characters.

```python

from itertools import permutations

def possible\_words(characters):

perms = [''.join(p) for p in permutations(characters)]

return perms

# Example usage

characters = "abc"

print(possible\_words(characters))

```

### 6. Python – Keys Associated with Values in Dictionary

To find keys associated with a particular value in a dictionary, you can use a list comprehension.

```python

def keys\_with\_value(d, value):

return [k for k, v in d.items() if v == value]

# Example usage

d = {'a': 1, 'b': 2, 'c': 1}

value = 1

print(keys\_with\_value(d, value))

```

### 7. Python Program to Find the Size of a Tuple

The size of a tuple can be found using the `len()` function.

```python

t = (1, 2, 3, 4)

size = len(t)

print(size)

```

### 8. Python – Maximum and Minimum K Elements in Tuple

To find the maximum and minimum `K` elements in a tuple, you can use sorting.

```python

def max\_min\_k\_elements(t, k):

sorted\_t = sorted(t)

return sorted\_t[:k], sorted\_t[-k:]

# Example usage

t = (1, 3, 5, 7, 9)

k = 2

print(max\_min\_k\_elements(t, k))

```

### 9. Create a List of Tuples from Given List Having Number and Its Cube in Each Tuple

You can use a list comprehension to create this list of tuples.

```python

numbers = [1, 2, 3, 4]

cubes = [(n, n\*\*3) for n in numbers]

print(cubes)

```

### 10. Python – Adding Tuple to List and Vice – Versa

To add a tuple to a list and vice versa:

```python

# Adding tuple to list

l = [1, 2, 3]

t = (4, 5)

l.extend(t)

print(l)

# Adding list to tuple

t = (1, 2, 3)

l = [4, 5]

t = t + tuple(l)

print(t)

```

### 11. Python – Closest Pair to Kth Index Element in Tuple

You can find the closest pair to a `K`th index element by calculating the absolute difference.

```python

def closest\_pair\_to\_k(t, k):

target = t[k]

closest\_pair = min(((t[i], t[j]) for i in range(len(t)) for j in range(i + 1, len(t))),

key=lambda x: abs(x[0] + x[1] - 2 \* target))

return closest\_pair

# Example usage

t = (1, 5, 9, 12, 15)

k = 2

print(closest\_pair\_to\_k(t, k))

```

Let me know if you need further details or any additional problems solved!