Here are 20 distinct dictionary problems for a deeper understanding and variety in dictionary handling in Python:

---

1. Create a Frequency Dictionary from List of Words

- **Description**: Write a function that accepts a list of words and returns a dictionary where keys are words and values are their frequencies.
- **Example**: `python word_frequency(["apple", "banana", "apple", "orange", "banana", "apple"])`
  - **Output**: `{'apple': 3, 'banana': 2, 'orange': 1}`

2. Dictionary Filter by Value

- **Description**: Write a function that filters a dictionary by only keeping entries where the value is above a certain threshold.
- **Example**: `python filter_by_value({'a': 5, 'b': 2, 'c': 8}, 4)`
  - **Output**: `{'a': 5, 'c': 8}`

3. Key with Maximum Value

- **Description**: Write a function that returns the key with the highest value in a dictionary.
- **Example**: `python max_key({'apple': 50, 'banana': 30, 'cherry': 20})`
  - **Output**: `'apple'`

4. Convert Dictionary to List of Tuples

- **Description**: Convert a dictionary into a list of tuples, each containing a key-value pair.
- **Example**: `python dict_to_tuples({'a': 1, 'b': 2, 'c': 3})`
  - **Output**: `[('a', 1), ('b', 2), ('c', 3)]`

5. Replace Dictionary Keys with Values from Another Dictionary

- **Description**: Given two dictionaries, replace keys in the first dictionary with their values from the second dictionary.
- **Example**: `python replace_keys({'x': 'a', 'y': 'b'}, {'a': 1, 'b': 2})`
  - **Output**: `{'x': 1, 'y': 2}`

6. Count Even and Odd Values in Dictionary

- **Description**: Given a dictionary of integer values, count how many values are even and how many are odd.
- **Example**: python `even_odd_count({'a': 1, 'b': 2, 'c': 3, 'd': 4})`
  - **Output**: `{'even': 2, 'odd': 2}`

7. Check if All Values are Unique

- **Description**: Write a function that checks if all values in a dictionary are unique.
- **Example**: python `unique_values({'a': 1, 'b': 2, 'c': 3})`
  - **Output**: `True`

8. Flatten a Multi-Level Nested Dictionary

- **Description**: Flatten a multi-level nested dictionary into a single dictionary with keys representing the path.
- **Example**: python `flatten({'a': {'b': {'c': 1}}, 'd': 2})`
  - **Output**: `{'a.b.c': 1, 'd': 2}`

9. Remove Key-Value Pairs with None Values

- **Description**: Write a function that removes any key-value pairs from a dictionary where the value is `None`.
- **Example**: python `remove_none({'a': 1, 'b': None, 'c': 2})`
  - **Output**: `{'a': 1, 'c': 2}`

10. Sort Dictionary by Values

- **Description**: Sort a dictionary by its values in descending order.
- **Example**: python `sort_by_value({'a': 3, 'b': 1, 'c': 2})`
  - **Output**: `{'a': 3, 'c': 2, 'b': 1}`

11. Merge Dictionaries with List Values

- **Description**: Merge multiple dictionaries where values are lists, concatenating lists for shared keys.
- **Example**: python `merge_list_values({'a': [1, 2]}, {'a': [3], 'b': [4]})`
  - **Output**: `{'a': [1, 2, 3], 'b': [4]}`

12. Remove Duplicate Values but Keep Keys

- **Description**: Write a function that removes duplicate values in a dictionary while keeping all keys intact.

- **Example**: python `remove_duplicates({'a': 1, 'b': 1, 'c': 2})`
  - **Output**: `{'a': 1, 'b': None, 'c': 2}`

13. Create a Multi-Level Dictionary from Lists

- **Description**: Create a multi-level dictionary from two lists, where one list represents the path and the other represents values.
- **Example**: python `create_multi_level(['a', 'b', 'c'], [1, 2, 3])`
  - **Output**: `{'a': {'b': {'c': 1}}}`

14. Find Common Keys in Multiple Dictionaries

- **Description**: Given a list of dictionaries, find the common keys present in all dictionaries.
- **Example**: python `common_keys([{'a': 1, 'b': 2}, {'b': 3, 'c': 4}])`
  - **Output**: `{'b'}`

15. Calculate Average of Values by Key Group

- **Description**: Given a dictionary where keys are grouped, calculate the average of values for each group.
- **Example**:
  python `average_by_key({'a': [1, 2, 3], 'b': [4, 5]})`
  - **Output**: `{'a': 2.0, 'b': 4.5}`

16. Convert String to Dictionary of Character Frequency

- **Description**: Given a string, convert it to a dictionary where each character is a key, and its frequency is the value.
- **Example**: python `char_freq("hello")`
  - **Output**: `{'h': 1, 'e': 1, 'l': 2, 'o': 1}`

17. Convert Dictionary Values to Percentages

- **Description**: Convert each value in a dictionary to a percentage of the total sum of values.
- **Example**: python `to_percentage({'a': 50, 'b': 50})`
  - **Output**: `{'a': 50.0, 'b': 50.0}`

18. Map Keys to List Elements

- **Description**: Given a dictionary of lists, map each key to a single value from each list to create new dictionaries.
- **Example**: python `map_keys({'a': [1, 2], 'b': [3, 4]})`
  - **Output**: `{'a': 1, 'b': 3}` (and so on for each combination)

19. Difference of Two Dictionaries by Value

- **Description**: Given two dictionaries, create a new dictionary showing the difference in values for matching keys.
- **Example**: python `diff_dict({'a': 5, 'b': 10}, {'a': 3, 'b': 8})`
  - **Output**: `{'a': 2, 'b': 2}`

20. Nested Dictionary Key Search

- **Description**: Write a function to search for a specific key in a nested dictionary and return the path to that key.
- **Example**: python `find_key({'a': {'b': {'c': 1}}}, 'c')`
  - **Output**: `'a.b.c'`

These problems cover a wide range of skills, from basic manipulations to more advanced dictionary operations. Let me know if you'd like further elaboration or code examples for any of these!