Python Data Structure

1. String & List: Given a sentence, split the sentence into words, remove duplicate words (maintaining the order), and return the result as a list.
2. List & Dictionary: Given a list of numbers, create a dictionary where each key is a number and the value is its frequency in the list.
3. Set & String: Write a Python program that takes two strings and returns the set of unique characters that appear in both strings.
4. Tuple & List: Given a list of tuples where each tuple contains two numbers, return a new list containing the product of the two numbers in each tuple.
5. String & Dictionary: Given a string, count the occurrences of each character and return the result as a dictionary.
6. List & Set: Write a Python program to find the common elements between two lists using a set for efficiency.
7. Dictionary & Tuple: Given a dictionary, return a sorted list of tuples where each tuple is a key-value pair from the dictionary, sorted by the dictionary's values.
8. List, Tuple, & Set: Given a list of tuples (where each tuple contains numbers), create a set containing the unique elements that appear across all the tuples.
9. Set & Dictionary: Write a Python program that checks if a dictionary has all unique values using a set.
10. String, List, & Dictionary: Given a paragraph, count the frequency of each word (case-insensitive) and return a dictionary where the keys are the words and the values are their frequencies.

11. Tuple & Set: Given a tuple of numbers, return a set of all even numbers from the tuple.
12. List & Dictionary: Write a Python program to group a list of words by their starting letter into a dictionary, where the key is the starting letter and the value is a list of words.
13. String, Tuple, & List: Given a string, convert it into a list of tuples where each tuple contains a character and its index in the string.
14. Set & List: Write a Python program that removes all elements from a list that appear more than once, preserving the original order.
15. List, Dictionary, & String: Given a list of sentences, create a dictionary where each key is a unique word and the value is the total number of sentences in which that word appears.
16. String, Set, & Dictionary: Given a string, create a dictionary where each key is a character, and the value is a set of indices where that character appears in the string.
17. List, Set & Tuple: Write a Python program to remove duplicates from a list of tuples (each tuple contains two elements), and return the result as a set of tuples.
18. String, List, & Set: Given a string, find all the unique words (case-insensitive) that appear in the string and return them as a sorted list.
19. Set & Dictionary: Given a dictionary of words as keys and their definitions as values, return a set containing all the words whose definitions contain a specific keyword.
20. List & Dictionary: Given a list of students with their names and grades (e.g., ("John", 85)), return a dictionary

where the keys are the grades and the values are lists of students who received that grade.

21. List, Set & Tuple: Given a list of tuples, where each tuple contains two integers, return a list of tuples where the sum of the integers in each tuple is unique.
22. String & Tuple: Write a program that takes a string, creates a tuple of each word in the string, and sorts the tuple in reverse alphabetical order.
23. Dictionary, List, & Set: Given a list of students and their grades, create a dictionary that groups students by their grade, and return a set of all unique grades.
24. String, Tuple, & Dictionary: Write a Python program that takes a string and returns a dictionary where the keys are words and the values are tuples, with the first element as the length of the word and the second as the count of vowels in the word.
25. Set, List, & String: Given two lists of strings, return a list of strings that are in both lists, but only include those that have more than 3 vowels.
26. Tuple, List, & Set: Given a list of tuples, each containing two integers, create a set of sums of all the pairs, and return the result as a sorted list.
27. String, List, & Dictionary: Write a program that takes a string and returns a dictionary where the keys are the first characters of words in the string, and the values are lists of words that start with that character.
28. List, Dictionary, & Set: Given a list of tuples (name, score), create a dictionary where the keys are the names, and the values are the scores, then return a set of names whose scores are above a certain threshold.

29. Tuple, List & Set: Write a Python program that takes a list of tuples, where each tuple contains a name and an age, and returns a set of unique ages, and creates a new list of tuples with the names of people who are older than 30.
30. String, Set, & List: Given a string of space-separated words, return a sorted list of words that appear more than once, without duplicates.
31. Tuple, Dictionary, & List: Given a list of tuples, where each tuple contains a product name and a price, create a dictionary where the keys are product names, and the values are the average prices of those products (in case of multiple occurrences).
32. Set, Dictionary, & List: Given a list of words, create a dictionary where the key is the word length, and the value is a set of words of that length.
33. List, Set, & String: Given a list of strings, create a set containing all unique characters across all strings, then return a list of these characters sorted alphabetically.
34. Dictionary, List, & String: Given a string of sentences, create a dictionary where the keys are the words (case-insensitive) and the values are lists of sentences in which each word appears.
35. Tuple, List & Set: Write a Python program that takes a list of tuples, where each tuple contains a name and an age, and returns a set of names whose age is greater than the average age of all people in the list.