Documentation for Group Anagrams

Overview:

The Group Anagrams problem involves grouping strings that are anagrams of each other together. An anagram is a word or phrase formed by rearranging the letters of another word or phrase, using all the original letters exactly once.

Problem Statement:

Given an array of strings `strs`, the task is to group the anagrams together. The order of the groups in the output is not significant.

Example:

- **Input:** strs : ["eat","tea","tan","ate","nat","bat"]
- **Output:** [["bat"],["nat","tan"],["ate","eat","tea"]]

Solution Approach:

The solution approach involves using a dictionary to store lists of anagrams. We iterate through each word in the input array. For each word, we sort its characters to create a unique representation. We then use this sorted representation as a key in the dictionary. We append the original word to the list corresponding to this key. Finally, we return the values of the dictionary, which are lists of anagrams.

Complexity Analysis:

- **Time Complexity:** (O(n \cdot k \log k)), where (n) is the length of the input array `strs`, and (k) is the maximum length of a string in `strs`. The time complexity is dominated by the sorting operation performed for each word.

- **Space Complexity:** (O(n \cdot k)), where (n) is the length of the input array `strs`, and (k) is the maximum length of a string in `strs`. This space is used to store the dictionary of anagrams.