[**Group Anagrams**](https://leetcode.com/problems/group-anagrams/)

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map.Entry;

**public** **class** GroupAnagrams {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String[] strs = {"eat", "tea", "tan", "ate", "nat", "bat"};

System.***out***.println(*groupAnagrams*(strs));

}

**public** **static** List<List<String>> groupAnagrams(String[] strs) {

List<List<String>> result = **new** ArrayList<>();

**if**(strs == **null** || strs.length == 0) {

**return** result;

}

HashMap<String , List<String>> map = **new** HashMap<>();

**for**(**int** i = 0 ; i < strs.length ; i++) {

**char**[] sorted = strs[i].toCharArray();

Arrays.*sort*(sorted);

String temp = **new** String(sorted);

**if**(!map.containsKey(temp)) {

map.put(temp , **new** ArrayList<>());

}

map.get(temp).add(strs[i]);

}

**for**(Entry<String, List<String>> entry : map.entrySet()) {

result.add(entry.getValue());

}

**return** result;

}

}

Time complexity : O(nlogk) , n is length of given array and k is average length of word

Space Complexity : O(nk) , n is length of given array and k is average length of word