public boolean isAnagram(String firstWord, String secondWord) {

char[] word1 = firstWord.replaceAll("[\\s]", "").toCharArray();

char[] word2 = secondWord.replaceAll("[\\s]", "").toCharArray();

Arrays.sort(word1);

Arrays.sort(word2);

return Arrays.equals(word1, word2);

}

|  |  |
| --- | --- |
|  | If you sort either array, the solution becomes O(n log n). but if you use a hashmap, it's O(n). tested and working.  char[] word1 = "test".toCharArray();  char[] word2 = "tes".toCharArray();  Map<Character, Integer> lettersInWord1 = new HashMap<Character, Integer>();  for (char c : word1) {  int count = 1;  if (lettersInWord1.containsKey(c)) {  count = lettersInWord1.get(c) + 1;  }  lettersInWord1.put(c, count);  }  for (char c : word2) {  int count = -1;  if (lettersInWord1.containsKey(c)) {  count = lettersInWord1.get(c) - 1;  }  lettersInWord1.put(c, count);  }  for (char c : lettersInWord1.keySet()) {  if (lettersInWord1.get(c) != 0) {  return false;  }  }  return true; |