1270. Ransom Note

Given an arbitrary ransom note string and another string containing letters from all the magazines, write a function that will return true if the ransom note can be constructed from the magazines ; otherwise, it will return false.

Each letter in the magazine string can only be used once in your ransom note.

Example

canConstruct("a", "b") -> false

canConstruct("aa", "ab") -> false

canConstruct("aa", "aab") -> true

Notice

You may assume that both strings contain only lowercase letters.

<https://www.lintcode.com/problem/ransom-note/description>

1. */\**
2. *\* To change this license header, choose License Headers in Project Properties.*
3. *\* To change this template file, choose Tools | Templates*
4. *\* and open the template in the editor.*
5. *\*/*
6. **package** javaapplication69;
8. **import** java.util.HashMap;
10. ***/\*\****
11. ***\****
12. ***\* @author Usuario***
13. ***\*/***
14. **public** **class** JavaApplication69 {
16. **public** **boolean** canConstruct(String ransomNote, String magazine) {
17. *// Write your code here*
18. *//return magazine.contains(ransomNote);*
19. HashMap<Character, Integer> hash\_ransom =
20. **new** HashMap();
22. **for**(**int** i =0; i<ransomNote.length(); i++)
23. {
24. **if**(hash\_ransom.containsKey(ransomNote.charAt(i)))
25. {
26. hash\_ransom.put(ransomNote.charAt(i), hash\_ransom.get(ransomNote.charAt(i)) + 1);
27. }
28. **else**
29. {
30. hash\_ransom.put(ransomNote.charAt(i),   1);
31. }
32. }
34. HashMap<Character, Integer> hash\_magazine =
35. **new** HashMap();
37. **for**(**int** i =0; i< magazine.length(); i++) {
38. **if**(  hash\_magazine.containsKey(magazine.charAt(i)) ) {
39. hash\_magazine.put( magazine.charAt(i) ,hash\_magazine.get( magazine.charAt(i) ) + 1);
40. }
41. **else**
42. {
43. hash\_magazine.put( magazine.charAt(i) ,  1);
44. }
45. }
47. **for**(**char** key : hash\_ransom.keySet()) {
48. **if**(!hash\_magazine.containsKey(key)) {
49. **if**(hash\_magazine.get(key) > hash\_ransom.get(key)) {
50. **return** **false**;
51. }
52. }
53. }
55. **return** **true**;
56. }
57. **public** **static** **void** main(String[] args) {
58. *// TODO code application logic here*
59. }
61. }