**[Verifying an Alien Dictionary](https://leetcode.com/problems/verifying-an-alien-dictionary/)**

**import** java.util.HashMap;

**public** **class** VerifyingAlienDictionary {

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

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

String[] words = {"apple","app"};

System.***out***.println(*isAlienSorted*(words , "abcdefghijklmnopqrstuvwxyz"));

}

**public** **static** **boolean** isAlienSorted(String[] words, String order) {

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

**return** **true**;

}

HashMap<Character , Integer> map = **new** HashMap<>();

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

map.put(order.charAt(i), i);

}

**for**(**int** i = 1 ; i < words.length ; i++) {

String first = words[i - 1];

String second = words[i];

**if**(!*isSorted*(map, first , second)) {

**return** **false**;

}

}

**return** **true**;

}

**public** **static** **boolean** isSorted(HashMap<Character , Integer> map , String first , String second) {

**for**(**int** i = 0 ; i < Math.*min*(first.length(), second.length()) ; i++) {

**char** c1 = first.charAt(i);

**char** c2 = second.charAt(i);

**if**(c1 != c2) {

**return** map.get(c1) <= map.get(c2);

}

}

**return** first.length() <= second.length();

}

}

Time Complexity : O(n \* l ) , n is length of words array and l is the average size of each word

Space Complexity : O(o), length of given order string