Hash Tables: Ransom Note

by saikiran9194

Check out the resources on the page's right side to learn more about hashing. The video tutorial is by Gayle Laakmann McDowell, author of the best-selling interview book Cracking the Coding Interview.

A kidnapper wrote a ransom note but is worried it will be traced back to him. He found a magazine and wants to know if he can cut out whole words from it and use them to create an untraceable replica of his ransom note. The words in his note are *case-sensitive* and he *must* use whole words available in the magazine, meaning he *cannot* use substrings or concatenation to create the words he needs.

Given the words in the magazine and the words in the ransom note, print `Yes` if he can replicate his ransom note *exactly* using whole words from the magazine; otherwise, print `No`.

Input Format

The first line contains two space-separated integers describing the respective values of m (the number of words in the magazine) and n (the number of words in the ransom note).
The second line contains m space-separated strings denoting the words present in the magazine.
The third line contains n space-separated strings denoting the words present in the ransom note.

Constraints

- $1 \leq m, n \leq 30000$
- $1 \leq \text{length of any word} \leq 5$.
- Each word consists of English alphabetic letters (i.e., a to z and A to Z).
- The words in the note and magazine are *case-sensitive*.

Output Format

Print `Yes` if he can use the magazine to create an untraceable replica of his ransom note; otherwise, print `No`.

Sample Input

```
6 4
give me one grand today night
give one grand today
```

Sample Output

```
Yes
```

Explanation

All four words needed to write an untraceable replica of the ransom note are present in the magazine, so we print `Yes` as our answer.

Submissions: 23127
Max Score: 25

Difficulty: Easy

Rate This Challenge:

Need Help?

6:25

Hashing

More

Current Buffer (saved locally, editable)

Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static boolean checkReplication(String[] magazine, String[] ransom) {

        boolean isValid = true;
        //populating HashMap "magazineMap" with magazine words
        Map<String, Integer> magazineMap = new HashMap<>();
        for (String word: magazine){
            if (magazineMap.containsKey(word) == false)
                magazineMap.put(word, 1);
            else
                magazineMap.put(word, magazineMap.get(word)+1);
        }

        //Checking for words of ransom note in magazineMap and deleting common ones, if uncommon set isValid = false

        for (String ransomNoteWord: ransom){
            if ( magazineMap.containsKey(ransomNoteWord) == false){
                isValid = false;
                break;
            } else {
                if( magazineMap.get(ransomNoteWord) == 0 ) {
                isValid = false;
                break;
            }
                magazineMap.put(ransomNoteWord, magazineMap.get(ransomNoteWord)-1);
            }
        }

        return isValid;
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int m = in.nextInt();
        int n = in.nextInt();
        String magazine[] = new String[m];
        for(int magazine_i=0; magazine_i < m; magazine_i++){
            magazine[magazine_i] = in.next();
```

```
        }
        String ransom[] = new String[n];
        for(int ransom_i=0; ransom_i < n; ransom_i++){
            ransom[ransom_i] = in.next();
        }

        boolean ans = checkReplication(magazine, ransom);
        if (ans == true)
            System.out.println("Yes");
        else
            System.out.println("No");
    }
}

```

Line: 30 Col: 36

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12
- ✔ Test Case #13
- ✔ Test Case #14
- ✔ Test Case #15
- ✔ Test Case #16
- ✔ Test Case #17
- ✔ Test Case #18
- ✔ Test Case #19

Next Challenge