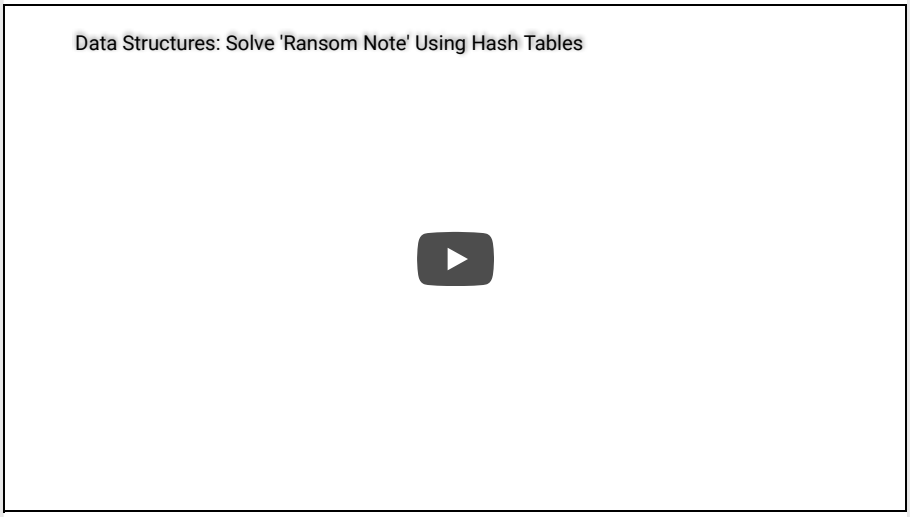
Hash Tables: Ransom Note

Video Explanation by gaylemcd



Editorial by AllisonP

We have a multiset of words in a magazine, $bagM$, and a multiset of words in a ransom note, $bagN$. If $bagN$ is contained in $bagM$, then we print `Yes` because we can recreate the note; otherwise, we print `No`.

Approach

To solve this challenge, we use two maps:

1. $mapM$, which maps each word in $bagM$ to its frequency.
2. $mapN$, which maps each word in $bagN$ to its frequency.

Because we only care that all the words in $bagN$ are contained in $bagM$, we simply iterate over $mapN$'s keyset and answer the following questions:

1. Does the key (from $mapN$'s keyset) exist in $mapM$?
2. If the key exists in both maps, is the value associated with that key in $mapN$ less than the value associated with the key in $mapM$?

If we answer no to the first question, we know that the magazine doesn't contain all the whole words we need to recreate the note. If we answer no to the second question, then we know that the word exists in the magazine but it simply does not occur enough times for us to recreate the entire note.

Set by saikiran9194

Problem Setter's code:

C++

```
#include <bits/stdc++.h>

using namespace std;

int main() {
    int m;              //number of words in magazine
    int n;              //number of words in notes
    cin >> m >> n;

    map<string, int> magazine; //map to store words in magazine with its frequency
```

STATISTICS

Difficulty: Easy
Time Complexity:
Required Knowledge: HashMaps
Publish Date: Sep 07 2016
This is a Practice Challenge

NEED HELP?

View discussions
View top submissions

```
    map<string, int> note;    //map to store words in notes with its frequency

    string word;

    for(int i = 0; i < m; i++) {
        cin >> word;
        magazine[word]++;
    }

    for(int i = 0; i < n; i++) {
        cin >> word;
        note[word]++;
    }

    map<string, int> :: iterator it;
    bool success = 1;

    //iterate over note map to check whether all words are present in map or not
    for(it = note.begin(); it != note.end(); it++) {
        if(magazine[it->first] < it->second) {
            success = 0;
            break;
        }
    }

    if(success) {
        cout << "Yes";
    }
    else {
        cout << "No";
    }

    return 0;
}
```

Tested by AllisonP

Problem Tester's code:

Java

```
import java.util.*;

public class Solution {
    Map<String, Integer> magazineMap;
    Map<String, Integer> noteMap;

    public Solution(String magazine, String note) {
        this.noteMap = new HashMap<String, Integer>();
        this.magazineMap = new HashMap<String, Integer>();

        // Must use an object instead of a primitive because it may be assigned a null refere
        Integer occurrences;

        for(String s : magazine.split("[^a-zA-Z]+")) {
            occurrences = magazineMap.get(s);

            if(occurrences == null) {
                magazineMap.put(s, 1);
            }
            else {
                magazineMap.put(s, occurrences + 1);
            }
        }

        for(String s : note.split("[^a-zA-Z]+")) {
            occurrences = noteMap.get(s);

            if(occurrences == null) {
                noteMap.put(s, 1);
            }
            else {
                noteMap.put(s, occurrences + 1);
            }
        }

    }

    public void solve() {
```

```
        boolean canReplicate = true;
        for(String s : noteMap.keySet()) {
            if(!magazineMap.containsKey(s) || (magazineMap.get(s) < noteMap.get(s)) ) {
                canReplicate = false;
                break;
            }
        }

        System.out.println( (canReplicate) ? "Yes" : "No" );
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int m = scanner.nextInt();
        int n = scanner.nextInt();

        // Eat whitespace to beginning of next line
        scanner.nextLine();

        Solution s = new Solution(scanner.nextLine(), scanner.nextLine());
        scanner.close();

        s.solve();
    }
}
```

Feedback

Was this editorial helpful?

Yes No