Interviewer B Packet - TIPS #5

# Interviewer:

## Behavioral:

## Question:

<https://leetcode.com/problems/jewels-and-stones/>

You're given strings jewels representing the types of stones that are jewels, and stones representing the stones you have. Each character in stones is a type of stone you have. You want to know how many of the stones you have are also jewels.

Letters are case sensitive, so "a" is considered a different type of stone from "A".

Examples:

Example 1:

Input: jewels = "aA", stones = "aAAbbbb"

Output: 3

Example 2:

Input: jewels = "z", stones = "ZZ"

Output: 0

## 

## Follow up Q&A:

Constraints:

* 1 <= jewels.length, stones.length <= 50
* jewels and stones consist of only English letters. (no numeric or special characters)
* All the characters of jewels are unique.

## Hint(s):

*Ask if they would like a hint before giving a hint*

## Solution(s): (General concept and time/space complexity)

#### Approach #1: Brute Force

Intuition and Algorithm

For each stone, check whether it matches any of the jewels. We can check with a linear scan.

class Solution {

int numJewelsInStones(string J, string S) {

int ans = 0;

for (char s: S) // For each stone...

for (char j: J) // For each jewel...

if (j == s) { // If the stone is a jewel...

ans++;

break; // Stop searching whether this stone 's' is a jewel

}

return ans;

}

}

Complexity Analysis

* Time Complexity:
* O(J.length \* S.length)
* Space Complexity:
  + O(1) (assuming they don’t create new arrays)

#### Approach #2: Hash Set

Intuition and Algorithm

For each stone, check whether it matches any of the jewels. We can check efficiently with a *Hash Set*.

class Solution {

int numJewelsInStones(String J, String S) {

Set<Character> Jset = new HashSet();

for (char j: J)

Jset.add(j);

int ans = 0;

for (char s: S)

if (Jset.contains(s))

ans++;

return ans;

}

}

Complexity Analysis

* Time Complexity:
  + *O*(*J*.length+*S*.length).
  + The *O*(*J*.length) part comes from creating Jset. The *O*(*S*.length) part comes from searching S.
* Space Complexity:
  + *O*(*J*.length).

### Other questions follow up

*Ask if there is more than 5 minutes remaining when they finish their code and testing.*

How would you change your solution if we limited the number of times a certain jewel can be counted to a maximum of k times?

For example,

Input: jewels = "ab", stones = "aAAbbbb", k = 3

Output: 4

# Interviewee:

## Question:

## Example(s):

## Code below or on leetcode