Kth Distinct String in an Array

## Description

A **distinct string** is a string that is present only **once** in an array.

Given an array of strings arr, and an integer k, return *the* kth ***distinct string*** *present in* arr. If there are **fewer** than k distinct strings, return *an **empty string*** "".

Note that the strings are considered in the **order in which they appear** in the array.

Example 1:

Input: arr = ["d","b","c","b","c","a"], k = 2

Output: "a"

Explanation:

The only distinct strings in arr are "d" and "a".

"d" appears 1st, so it is the 1st distinct string.

"a" appears 2nd, so it is the 2nd distinct string.

Since k == 2, "a" is returned.

Example 2:

Input: arr = ["aaa","aa","a"], k = 1

Output: "aaa"

Explanation:

All strings in arr are distinct, so the 1st string "aaa" is returned.

Example 3:

Input: arr = ["a","b","a"], k = 3

Output: ""

Explanation:

The only distinct string is "b". Since there are fewer than 3 distinct strings, we return an empty string "".

## Algorithm

1.) **Count Frequency of Each String**:

- Use a HashMap called freqMap to store the frequency of each string in the arr array. Iterate through the array and populate the freqMap with the count of each string.

2.) **Identify the k-th Distinct String**:

- Iterate through the arr array again. For each string, check if its frequency in freqMap is 1 (i.e., it is distinct).
- If a string is distinct, decrement k.
- When k reaches 0, return the current string as it is the k-th distinct string.

3.) **Handle Case Where k-th Distinct String is Not Found**:

- If the loop completes without finding the k-th distinct string, return an empty string "".

## Pseudocode

```
function kthDistinct(arr: array of String, k: int) -> String:
  freqMap = new HashMap<String, Integer>()

  # Count frequency of each string
  for each str in arr:
    freqMap.put(str, freqMap.getOrDefault(str, 0) + 1)

  # Find the k-th distinct string
  for each str in arr:
    if freqMap.get(str) == 1:
      k -= 1
    if k == 0:
      return str

  # Return empty string if k-th distinct string is not found
  return ""
```

## Code

```
class Solution {
    public String kthDistinct(String[] arr, int k) {
        HashMap<String, Integer> freqMap = new HashMap<>();
        for(String str: arr){
            freqMap.put(str, freqMap.getOrDefault(str, 0) + 1);
        }
        for(String str: arr){
            if(freqMap.get(str)==1){
                k--;
            }
            if (k == 0){
                return str;
            }
        }
        return "";
    }
}
```

## Conclusion

The kthDistinct function finds the k-th distinct string in an array by first counting the frequency of each string using a HashMap and then iterating through the array again to identify the k-th distinct string. If the k-th distinct string is found, it is returned; otherwise, an empty string is returned. This approach ensures that the function efficiently handles the identification of distinct strings while maintaining clarity and correctness.