Fruits into Baskets (medium)

We'll cover the following

Problem Statement

You are visiting a farm to collect fruits. The farm has a single row of fruit trees. You will be given two baskets, and your goal is to pick as many fruits as possible to be placed in the given baskets.

You will be given an array of characters where each character represents a fruit tree. The farm has following restrictions:

1. Each basket can have only one type of fruit. There is no limit to how many fruit a basket can hold.
2. You can start with any tree, but you can’t skip a tree once you have started.
3. You will pick exactly one fruit from every tree until you cannot, i.e., y will stop when you have to pick from a third fruit type.

Write a function to return the maximum number of fruits in both baskets.

Example 1:

```
Input: Fruit=['A', 'B', 'C', 'A', 'C']
Output: 3
Explanation: We can put 2 'C' in one basket and one 'A' in the other fro
m the subarray ['C', 'A', 'C']
```

Example 2:

```
Input: Fruit=['A', 'B', 'C', 'B', 'B', 'C']
Output: 5
Explanation: We can put 3 'B' in one basket and two 'C' in the other bas
ket.
This can be done if we start with the second letter: ['B', 'C', 'B', 'B'
, 'C']
```

Try it yourself

Try solving this question here:

C++

```
using namespace std;

#include <iostream>
#include <unordered_map>
#include <vector>

class MaxFruitCountOf2Types {
 public:
  static int findLength(const vector<char>& arr) {
    int maxLength = 0;
    // TODO: Write your code here
    return maxLength;
  }
};
```

Solution

This problem follows the **Sliding Window** pattern and is quite similar to Longest Substring with K Distinct Characters. In this problem, we need to find the length of the longest subarray with no more than two distinct characters (or fruit types!). This transforms the current problem into **Longest Substring with K Distinct Characters** where K=2.

Code

Here is what our algorithm will look like, only the highlighted lines are different from Longest Substring with K Distinct Characters:

C++

```
using namespace std;

#include <iostream>
#include <unordered_map>
#include <vector>

class MaxFruitCountOf2Types {
 public:
  static int findLength(const vector<char>& arr) {
    int windowStart = 0, maxLength = 0;
    unordered_map<char, int> fruitFrequencyMap;
    // try to extend the range [windowStart, windowEnd]
    for (int windowEnd = 0; windowEnd < arr.size(); windowEnd++) {
      fruitFrequencyMap[arr[windowEnd]]++;
      // shrink the sliding window, until we are left with '2' fruits in the
      while ((int)fruitFrequencyMap.size() > 2) {
        fruitFrequencyMap[arr[windowStart]]--;
        if (fruitFrequencyMap[arr[windowStart]] == 0) {
          fruitFrequencyMap.erase(arr[windowStart]);
        }
        windowStart++;  // shrink the window
      }
      maxLength = max(maxLength, windowEnd - windowStart + 1);
    }

    return maxLength;
  }
```

Time Complexity

The above algorithm’s time complexity will be $O(N)$, where ‘N’ is the number of characters in the input array. The outer `for` loop runs for all characters, and the inner `while` loop processes each character only once; therefore, the time complexity of the algorithm will be $O(N+N)$, which is asymptotically equivalent to $O(N)$.

Space Complexity

The algorithm runs in constant space $O(1)$ as there can be a maximum of three types of fruits stored in the frequency map.

Similar Problems

Problem 1: Longest Substring with at most 2 distinct characters

Given a string, find the length of the longest substring in it with at most two distinct characters.

Solution: This problem is exactly similar to our parent problem.

← Back

Longest Substring with maximum K D...

Longest Substring with Distinct Chara...

☑ Mark as Completed

Report an Issue