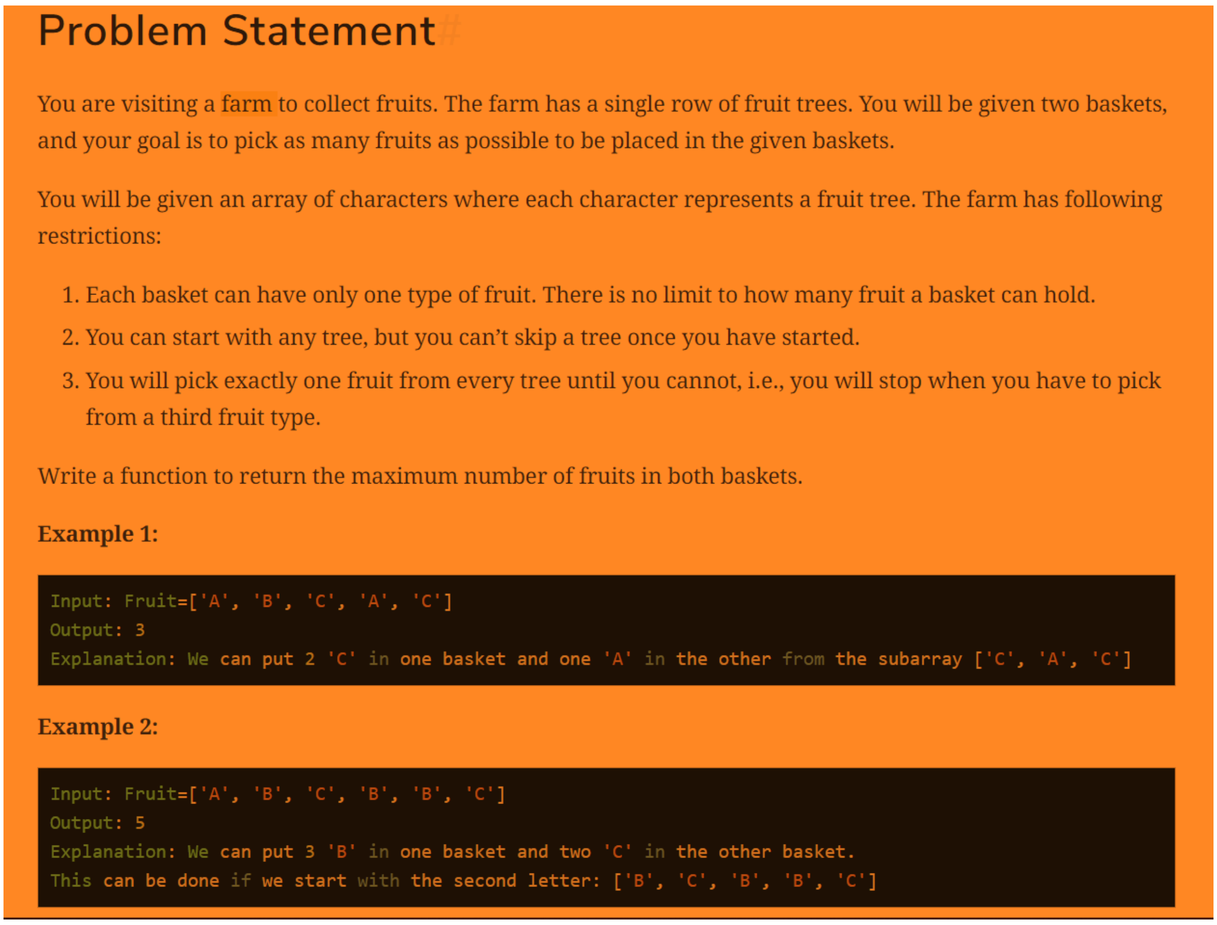
Fruits into Baskets

SR Score	1002
Link	https://www.educative.io/courses/grokking-the-coding-interview/Bn2KLlOR0lQ
Last Reviewed	@April 8, 2022
Time	5
Score	4
DS	arrays
Algo	sliding window
Stated	medium
Perceived	easy
List	REPEAT
Needs Review	☑
Repeat Offender	□
Confident	□
C_Date	1
C_Solution	2
C_Time	500
Frequency	

▼ **Problem Statement**



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., you 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 from 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 basket.
This can be done if we start with the second letter: ['B', 'C', 'B', 'B', 'C']
```

▼ **Intuition**

- key is translating each part of the fruits & farms story to it's more boring LC parallel
- ▼ You are visiting a to collect fruits
  - jk!
- ▼ The farm has a single row of fruit trees.
  - there is an array of characters
- ▼ 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 2 “character slots”, and your goal is to find the maximum # of characters as possible that fit within the 2 character slots

  - ▼ 1. Each basket can have only one type of fruit. There is no limit to how many fruit a basket can hold.
    - each character slot can only have one letter of the alphabet, but there is no limit to the # of that letter of the alphabet the slot can hold
  - ▼ 2. You can start with any tree, but you can’t skip a tree once you have started.
    - you can start with any letter, but cannot skip a letter once you’ve started
  - ▼ 3. You will pick exactly one fruit from every tree until you cannot, i.e., you will stop when you have to pick from a third fruit type.
    - You will pick exactly one letter from each index in the array until you cannot, i.e. you will stop when you have to pick a third unique letter from the alphabet
  - ▼ Write a function to return the maximum number of fruits in both baskets.
    - Write a function to return the maximum length substring containing 2 distinct characters

▼ **Time & Space Considerations**

- Time: O(2n) → O(n)
  - for loop to iterate thru all characters → O(n)
  - while loop inside for loop processes each letter in array 1 time → O(n)
  - $O(n + n) = O(2n) - - > O(n)$
- Space: O(3) → O(1)
  - dictionary $fruitTracker$ can hold a max of 3 elements (keys) since problem specifies we only get 2 “fruit baskets” (aka distinct characters), so size of dict is constant always

▼ **Review Notes**

- ▼ [3/5/22]
  - didn’t have a clue, checked solution
- ▼ [4/8/22]
  - got code optimally in sub 5 min
  - key growth area from [3/5/22] is that you could find parallels between the farm + fruits story and the actual LC problem it’s asking you

▼ **Tracking**

Scores

Aa Attempt #	Date	# Time	# Score
2	@April 8, 2022	5	5
1	@March 5, 2022	1	1
Untitled			

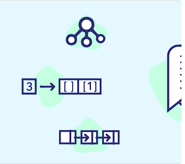
▼ **Solutions**

```
# solve 2: 4/8/22 (optimal first try, sub 5 min)
#time: O(2n) --> O(n)
#space: O(3) --> O(1)
def fruits_into_baskets(fruits):
    fruitTracker = dict()
    maxNumFruits = windowStart = 0
    for windowEnd in range(len(fruits)):
        fruit = fruits[windowEnd]
        fruitTracker[fruit] = fruitTracker.get(fruit, 0) + 1

        while len(fruitTracker) > 2:
            leftFruit = fruits[windowStart]
            fruitTracker[leftFruit] -= 1
            if fruitTracker[leftFruit] == 0:
                del fruitTracker[leftFruit]
            windowStart += 1

        maxNumFruits = max(maxNumFruits, windowEnd - windowStart + 1)

    return maxNumFruits


def main():
  print("Maximum number of fruits: " + str(fruits_into_baskets(['A', 'B', 'C', 'A', 'C'])))
  print("Maximum number of fruits: " + str(fruits_into_baskets(['A', 'B', 'C', 'B', 'B', 'C'])))

main()
# ----------------------------------------------------------------------------------
# solve 1: 3/5 (looked at solution)
def fruits_into_baskets(fruits):
    maxFruits = windowStart = 0
    fruitCounts = dict()
    for windowEnd in range(len(fruits)):
        fruit = fruits[windowEnd]
        fruitCounts[fruit] = fruitCounts.get(fruit, 0) + 1
        while len(fruitCounts) > 2:
            leftFruit = fruits[windowStart]
            fruitCounts[leftFruit] -= 1
            if fruitCounts[leftFruit] == 0:
                del fruitCounts[leftFruit]
            windowStart += 1
        maxFruits = max(maxFruits, windowEnd - windowStart + 1)

    return maxFruits

def main():
  print("Maximum number of fruits: " + str(fruits_into_baskets(['A', 'B', 'C', 'A', 'C'])))
  print("Maximum number of fruits: " + str(fruits_into_baskets(['A', 'B', 'C', 'B', 'B', 'C'])))

main()
```

▼ Resources

Fruits into Baskets (medium) - Grokking the Coding Interview: Patterns for Coding Questions

https://www.educative.io/courses/grokking-the-coding-interview/Bn2KLlOR0lQ



Longest Substring with maximum K Distinct Characters (medium) - Grokking the Coding Interview: Patterns for Co

https://www.educative.io/courses/grokking-the-coding-interview/YQQwQMWLx80

▼ GitHub

GCI/Pattern 1 - Sliding Window/Fruits into Baskets at main · psdev30/GCI

Contribute to psdev30/GCI development by creating an account on GitHub.

https://github.com/psdev30/GCI/tree/main/Pattern%201%20-%20Sliding%20Window/Fruits%20into%20Baskets