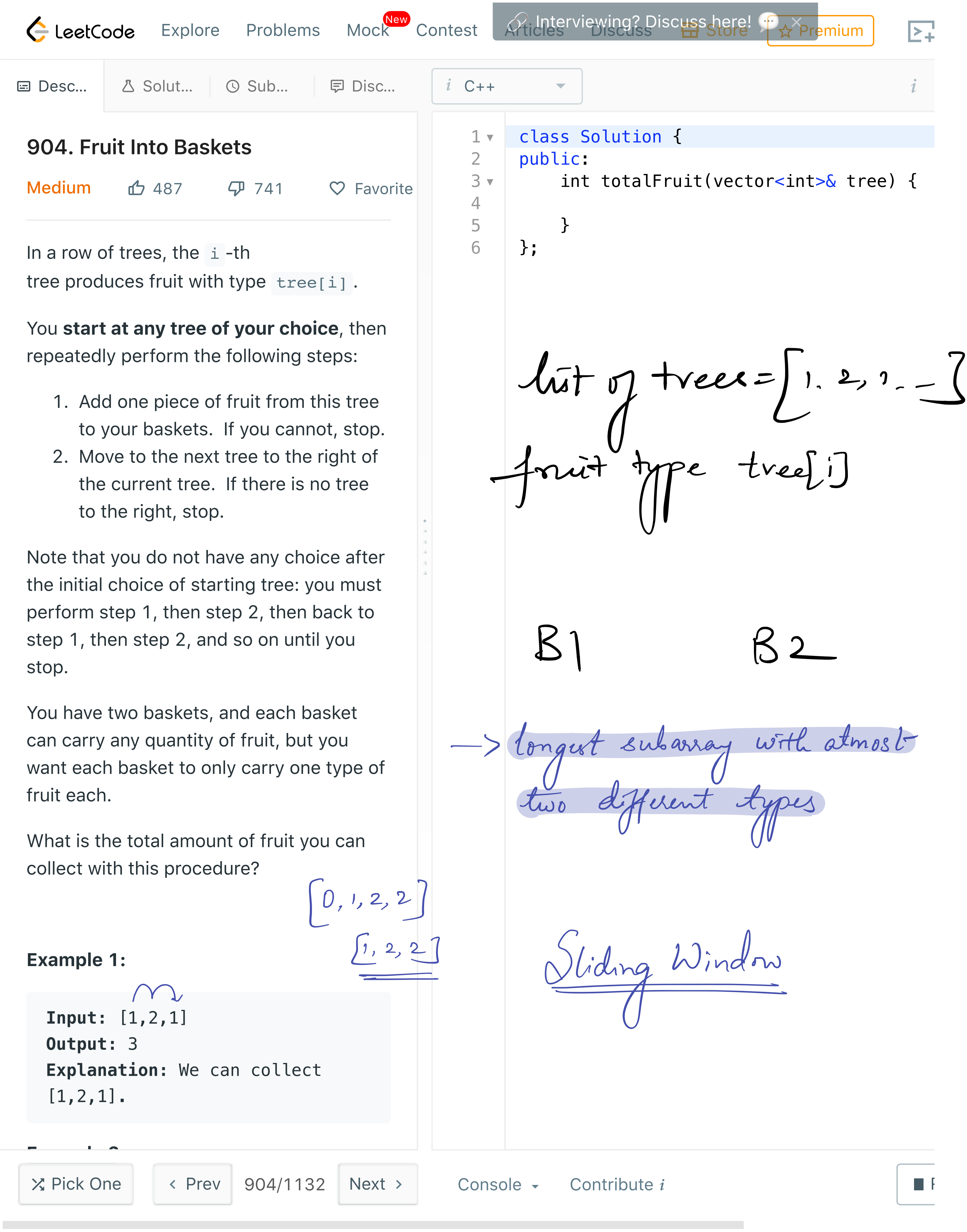


904. Fruit Into Baskets

Medium 487 741 Favorite

In a row of trees, the `i` -th tree produces fruit with type `tree[i]`.

You **start at any tree of your choice**, then repeatedly perform the following steps:

1. Add one piece of fruit from this tree to your baskets. If you cannot, stop.
2. Move to the next tree to the right of the current tree. If there is no tree to the right, stop.

Note that you do not have any choice after the initial choice of starting tree: you must perform step 1, then step 2, then back to step 1, then step 2, and so on until you stop.

You have two baskets, and each basket can carry any quantity of fruit, but you want each basket to only carry one type of fruit each.

What is the total amount of fruit you can collect with this procedure?

[0, 1, 2, 2]

[1, 2, 2]

Example 1:

```
Input: [1,2,1]
Output: 3
Explanation: We can collect
[1,2,1].
```

```cpp
class Solution {
public:
    int totalFruit(vector<int>& tree) {
        
    }
};
```

list of trees = [1, 2, 1, _ _]

fruit type tree[i]

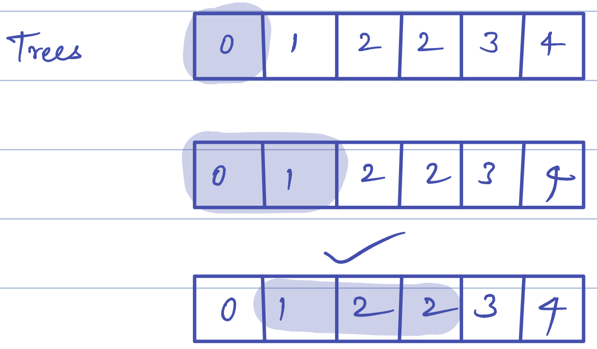
B1 B2

→ longest subarray with atmost two different types

Sliding Window

Explanation

Types of fruits = [0, 1, 2, 3, 4]



Trees | 0 | 1 | 2 | 2 | 3 | 4

0 | 1 | 2 | 2 | 3 | 4 → 2 different trees for 2 fruits

0 | 1 | 2 | 2 | 3 | 4 → 3 trees for 2 diff fruits

output: No of trees you can visit so that you collect 2 different fruits

Algorithm :

{fruit 1: num,
fruit 2: num.....}

```
1. Use a hash_map
2. Add diff fruits to the hash map
3. while the no. of fruits > 2
       decrement the no. of fruits for the first type
     if num[fruit type 1] == 0
         delete the fruit type
     Move the window to the right
4. Max(max_len, win_end - win_start + 1)
```