Find Missing And Repeating

Given an unsorted array **Arr** of size **N** of positive integers. **One number 'A'** from set {1, 2, …N} is missing and **one number 'B'** occurs twice in array. Find these two numbers.

Example 1:

Input: N = 2
Arr[] = {2, 2}
Output: 2 1
Explanation: Repeating number is 2 and
smallest positive missing number is 1.

Example 2:

Input: N = 3
Arr[] = {1, 3, 3}
Output: 3 2
Explanation: Repeating number is 3 and
smallest positive missing number is 2.

Your Task:
You don't need to read input or print anything. Your task is to complete the function **findTwoElement()** which takes the array of integers **arr** and **n **as parameters and returns an array of integers of size 2 denoting the answer (The first index contains **B **and second index contains **A.)**

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

This is very beautiful question. My approach is using bit manipulation and using space.

```
def findTwoElement( self,arr, n):
        # code here
        res = [False]*(n+1)
        dup = 0
        miss = 0
        for ele in arr:
            if res[ele]==False:
                res[ele]=True
            elif res[ele]==True:
                dup = ele
        for i in range(1,len(res)):
            if res[i] is False:
                miss = i
        return dup,miss
```

Better method using constant space:

```
def printTwoElements( arr, size):
    for i in range(size):
        if arr[abs(arr[i])-1] > 0:
            arr[abs(arr[i])-1] = -arr[abs(arr[i])-1]
        else:
            print("The repeating element is", abs(arr[i]))

    for i in range(size):
        if arr[i]>0:
            print("and the missing element is", i + 1)

# Driver program to test above function */
arr = [7, 3, 4, 5, 5, 6, 2]
n = len(arr)
printTwoElements(arr, n)
```