

**3.Hard\09.Repeating\_and\_missing\_numbers.cpp**

```

1  /*
2  QUESTION:
3  Given an unsorted array Arr of size N of positive integers. One number 'A' from set {1,
4  2,...,N} is missing and one number 'B' occurs twice in the array. Find these two numbers.
5
6  Example:
7  Input:
8  N = 2
9  Arr[] = {2, 2}
10 Output: 2 1
11 Explanation: Repeating number is 2 and the smallest positive missing number is 1.
12
13 APPROACH:
14 To find the missing and repeating numbers in the given unsorted array, we can utilize the
15 properties of summation and sum of squares. Let's denote the missing number as 'x' and the
16 repeating number as 'y'.
17
18 1. Calculate the optimal sum 'optSum' using the formula:  $\text{optSum} = N * (N + 1) / 2$ , where N is
19 the size of the array.
20 2. Calculate the optimal sum of squares 'opt2Sum' using the formula:  $\text{opt2Sum} = N * (N + 1) * (2 * N + 1) / 6$ .
21 3. Calculate the actual sum 'actSum' and actual sum of squares 'act2Sum' of the given array.
22 4. Find the difference between the optimal sum and the actual sum:  $x\text{MinusY} = \text{optSum} - \text{actSum}$ .
23 5. Find the difference between the optimal sum of squares and the actual sum of squares:
24  $x2\text{MinusY2} = \text{opt2Sum} - \text{act2Sum}$ .
25 6. Calculate the sum of 'x' and 'y':  $x\text{PlusY} = x2\text{MinusY2} / x\text{MinusY}$ .
26 7. Calculate 'x' and 'y' using the equations:  $x = (x\text{PlusY} + x\text{MinusY}) / 2$  and  $y = x\text{PlusY} - x$ .
27
28 CODE:
29 */
30
31 vector<int> findTwoElement(vector<int> arr, int N) {
32     long long n = N;
33     long long optSum = n * (n + 1) / 2; // Sum if all elements are present once
34     long long opt2Sum = n * (n + 1) * (2 * n + 1) / 6; // Optimum sum of squares
35     long long actSum = 0; // Actual sum of the given array
36     long long act2Sum = 0; // Actual sum of squares
37
38     for (auto it : arr) {
39         actSum += it;
40         act2Sum += (long long)it * (long long)it;
41     }
42
43     long long xMinusY = optSum - actSum;
44     long long x2MinusY2 = opt2Sum - act2Sum;
45     long long xPlusY = x2MinusY2 / xMinusY;
46
47     long long x = (xPlusY + xMinusY) / 2;
48     long long y = xPlusY - x;
49
50     return {(int)y, (int)x};
51 }

```

```
47  
48  /*  
49  TIME COMPLEXITY: O(N), where N is the size of the array.  
50  SPACE COMPLEXITY: O(1).  
51  */  
52
```