Check out Codility training tasks

Candidate Report: Anonymous

Test Name:

Summary Timeline

Test Score

50 out of 100 points

50%

Tasks in Test

	Time Spent	Task Score
PermMissingElem Submitted in: Java 8	16 min	50%

TASKS DETAILS

		Task Score	Correctness	Performance
EASY	1. PermMissingElem Find the missing element in a given permutation.	50%	100%	0%

Task description

An array A consisting of N different integers is given. The array contains integers in the range [1..(N + 1)], which means that exactly one element is missing.

Your goal is to find that missing element.

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given an array A, returns the value of the missing element.

For example, given array A such that:

```
A[0] = 2
A[1] = 3
A[2] = 1
A[3] = 5
```

the function should return 4, as it is the missing element.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..100,000];
- the elements of A are all distinct;
- each element of array A is an integer within the range [1..(N + 1)].

Solution

Programming language used:	Java 8
Total time used:	17 minutes
Effective time used:	16 minutes
Notes:	*not defined yet*

Task timeline

09:44:14 10:00:16

Code: 10:00:15 UTC, java, final, score: **50**

show code in pop-up

```
// you can also use imports, for example:
// import java.util.*;

// you can write to stdout for debugging purposes, e.g.
// System.out.println("this is a debug message");

class Solution {
        public static int max = 10;
        public static boolean[] mem;


        public static int solution(int[] a) {
                mem = new boolean[max + 1];
                if (a.length == 0)
                        return 1;

                for (int i = 0; i < a.length; i++) {
                        mem[a[i]] = true;
                }

//              System.out.println(Arrays.toString(a));
//              System.out.println(Arrays.toString(mem));

                for (int i = 1; i < a.length + 1; i++) {
                        if (!mem[i])
                                return i;
                }

                return a.length + 1;
        }
}
```

Analysis summary

The following issues have been detected: runtime errors.

Analysis

Detected time complexity: **O(N ** 2)**

expand all **Example tests**

▶ example example test	✓ OK

expand all **Correctness tests**

▶ empty_and_single empty list and single element	✓ OK
▶ missing_first_or_last the first or the last element is missing	✓ OK
▶ single single element	✓ OK
▶ double two elements	✓ OK
▶ simple simple test	✓ OK

expand all **Performance tests**

▶ medium1 medium test, length = ~10,000	✗ RUNTIME ERROR tested program terminated with exit code 1
▶ medium2 medium test, length = ~10,000	✗ RUNTIME ERROR tested program terminated with exit code 1
▶ large_range range sequence, length = ~100,000	✗ RUNTIME ERROR tested program terminated with exit code 1
▶ large1 large test, length = ~100,000	✗ RUNTIME ERROR tested program terminated with exit code 1
▶ large2 large test, length = ~100,000	✗ RUNTIME ERROR tested program terminated with exit code 1