

Problem 1. Isolation

Nodde has a new problem¹, he has some bottles of chemicals lined up on a shelf in some order, and the issue is that chemicals of the same type tend to end up being too volatile if kept together. So he intends to put some dividers between the bottles to isolate them. He's curious to find out the largest possible contiguous interval in which all bottles are unique. Each bottle will be labelled with a non-zero integer for a type. Your solution should run in expected $O(n)$ time, where n is the number of bottles.

For example, consider the following arrangement of bottles:

[1, 2, 20, 4, 3, 2, 7, 9]

Then the largest possible interval is 6, since the two bottles of type 2 need to be separated, so the largest interval is given as: [20, 4, 3, 2, 7, 9].

Another example is the following”

[4, 5, 4, 5, 4, 5]

Then the largest possible interval is 2.

```
HashMap<Integer, Integer> map: element, index
int startIdx = 0;
int maxLen = 0;

for (int i = 0; i < arr.length; i++) {
    if (map.containsKey(arr[i])) {
        startIdx = Math.max(map.get(arr[i]) + 1, startIdx);
    }
    map.put(arr[i], i);

    maxLen = Math.max(i - startIdx + 1, maxLen);
}

return maxLen;
```

¹Quite troublesome if you think about it, every week he has some other conundrum that needs solving.