**15. ValueOccurrences**

**Given a sorted array, calculate how many insertion and removal operations are required to make every value X in the array occur exactly X times.**

**Task description:**

* There is an array A of N integers sorted in non-decreasing order. In one move, you can either remove an integer from A or insert an integer before or after any element of A. The goal is to achieve an array in which all values X that are present in the array occur exactly X times.
* **For example,** given A = [1, 1, 3, 4, 4, 4], value 1 occurs twice, value 3 occurs once and value 4 occurs three times. You can remove one occurrence each of both 1 and 3, and insert one occurrence 4, resulting in the array [1, 4, 4, 4, 4]. In this array, every element X occurs exactly X times.
* What is the minimum number of moves after which every value X in the array occurs exactly X times?
* Write a function: **function solution(A);** that, given an array A, returns **the minimum number of moves** after which every value X in the array occurs exactly X times. Note that it is permissible to remove some values entirely, if appropriate.
* **Examples:**
* Given A = [1, 1, 3, 4, 4, 4], your function should **return 3,** as described above.
* Given A= [1, 2, 2, 2, 5, 5, 5, 8], your function should **return 4.** You can delete the 8 and one occurrence of 2, and insert 5 twice, resulting in [1, 2, 2, 5, 5, 5, 5, 5] after four moves. Notice that after the removals, there is no occurrence of 8 in the array anymore.
* Given A [1, 1, 1, 1, 3, 3, 4, 4, 4, 4, 4], your function should **return 5**
* Given A [10, 10, 10], your function should **return 3.** You can remove all elements, resulting in an empty array.
* **Write an efficient algorithm for the following assumptions:**
* N is an integer within the range [1..100,000];
* each element of array A is an integer within the range [1..100,000,000];
* elements of array A are sorted in non-decreasing order.