# AE2ADS

# Reading

M. T. Goodrich, R. Tamassia and M. H. Goldwasser, *Data Structures and Algorithms in Java*, 6th Edition, 2014.

- Chapter 10. Hash Tables, Maps and Skip Lists

# Exercise 1

Implement the Map ADT using an unsorted list. Analyze the time complexity of the implemented methods.

# Exercise 2

Implement the Map ADT using a hash table. If any collision occurs, handle it using “Separate Chaining”.

# Exercise 3

Given an integer array $nums$ and an integer target value $target$, please find the two integers in the array whose sum is the target value $target$, and return their array indices.

You can assume that each input will only correspond to one answer, and you cannot use the same element twice.

Example 1:

- Input: nums = [2,7,11,15], target = 9
- Output: [0, 1]

Example 2:

- Input: nums = [3,2,4], target = 6
- Output: [1, 2]

Example 3:

- Input: nums = [3,3], target = 6
- Output: [0, 1]

# Exercise 4

Given an array nums of size $n$, return the majority element. A majority element is an element that occurs more than $\left\lfloor \frac{n}{2} \right\rfloor$ times in the array.

You may assume that the array is non-empty and that there is always a majority element in the given array.

Example 1

- Input: nums = [3, 2, 3]
- Output: 3

Example 2

- Input: nums = [2,2,1,1,1,2,2]
- Output: 2

# Exercise 5 (ICPC)

A new e-mail service "Berlandesk" is going to be opened in Berland in the near future. The site administration wants to launch their project as soon as possible, that's why they ask you to help. You're suggested to implement the prototype of site registration system. The system should work on the following principle.

Each time a new user wants to register, he sends to the system a request with his name. If such a name does not exist in the system database, it is inserted into the database, and the user gets the response OK, confirming the successful registration. If the name already exists in the system database, the system makes up a new user name, sends it to the user as a prompt and also inserts the prompt into the database. The new name is formed by the following rule. Numbers, starting with 1, are appended one after another to name (name1, name2, ...), among these numbers the least $i$ is found so that name $i$ does not yet exist in the database.

https://www.luogu.com.cn/problem/CF4C

# Exercise 6 (ICPC)

Polycarp likes to play with numbers. He takes some integer number $x$ , writes it down on the board, and then performs with it $n-1$ operations of the two kinds:

- divide the number $x$ by $3$ ( $x$ must be divisible by $3$ );
- multiply the number $x$ by 2 .

After each operation, Polycarp writes down the result on the board and replaces $x$ by the result. So there will be $n$ numbers on the board after all.

You are given a sequence of length $n$ — the numbers that Polycarp wrote down. This sequence is given in arbitrary order, i.e., the order of the sequence can mismatch the order of the numbers written on the board.

Your problem is to rearrange (reorder) elements of this sequence in such a way that it can match possible Polycarp's game in the order of the numbers written on the board. I.e., each next number will be exactly two times of the previous number or exactly one third of previous number.

It is guaranteed that the answer exists.

https://www.luogu.com.cn/problem/CF977D

# More ICPC Exercises

https://www.luogu.com.cn/problem/list?type=CF&page=1&tag=235