Problem Description

Given an integer array nums sorted in **non-decreasing order**, remove the duplicates  such that each unique element appears only **once** and return the new array.

**Example 1:**

**Input:** nums = [1,1,2]

**Output:** [1,2]

**Explanation:** Array after removing duplicates are [1,2]

**Constraints:**

* 1 <= nums.length <= 3∗1043∗104
* -100 <= nums[i] <= 100
* nums is sorted in **non-decreasing** order.

Solution:

1. Method by using list of Integer: (List<Integer>)

class Solution {

    public List<Integer> removeDuplicates(List<Integer> nums) {

        // Your code goes here

        List<Integer> original = new ArrayList<Integer>();

        int traverse=0;

        while(traverse<nums.size()-1)

        {

            int startNum=nums.get(traverse);

            int traverseNum=nums.get(traverse+1);

            if(startNum!=traverseNum)

                original.add(nums.get(traverse));

            traverse++;

        }

        original.add(nums.get(traverse));

        return original;

    }

}

1. Method by using array of integer:

Will do soon.