1. Find the Majority Element in an Array

Import java.util.\*;

Public class MajorityElement {

Public static int findMajorityElement(int[] nums) {

Int count = 0, candidate = 0;

For (int num : nums) {

If (count == 0) {

Candidate = num;

}

Count += (num == candidate) ? 1 : -1;

}

Return candidate;

}

Public static void main(String[] args) {

Int[] nums = {3, 3, 4, 2, 3, 3, 3};

System.out.println(findMajorityElement(nums)); // Output: 3

}

}

1. Maximum Subarray Sum (Kadane’s Algorithm)

Public class MaximumSubarraySum {

Public static int maxSubArray(int[] nums) {

Int maxSoFar = nums[0], maxEndingHere = nums[0];

For (int I = 1; I < nums.length; i++) {

maxEndingHere = Math.max(nums[i], maxEndingHere + nums[i]);

maxSoFar = Math.max(maxSoFar, maxEndingHere);

}

Return maxSoFar;

}

Public static void main(String[] args) {

Int[] nums = {-2, 1, -3, 4, -1, 2, 1, -5, 4};

System.out.println(maxSubArray(nums)); // Output: 6

}

}

1. First Non-Repeating Character in a String

Import java.util.\*;

Public class FirstUniqueCharacter {

Public static int firstUniqChar(String s) {

Int[] freq = new int[26];

For (char c : s.toCharArray()) {

Freq[c – ‘a’]++;

}

For (int I = 0; I < s.length(); i++) {

If (freq[s.charAt(i) – ‘a’] == 1) return I;

}

Return -1;

}

Public static void main(String[] args) {

System.out.println(firstUniqChar(“mountain”)); // Output: 0

System.out.println(firstUniqChar(“aabb”)); // Output: -1

}

}

1. Check if One String is a Rotation of Another

Public class StringRotation {

Public static boolean isRotation(String s1, String s2) {

If (s1.length() != s2.length()) return false;

Return (s1 + s1).contains(s2);

}

Public static void main(String[] args) {

System.out.println(isRotation(“waterbottle”, “erbottlewat”)); // True

System.out.println(isRotation(“hello”, “lohel”)); // True

System.out.println(isRotation(“abc”, “acb”)); // False

}

}