Naga Rajasekhar

Arrays - 1st Assignment

**1st Solution**

function twoSum(nums, target) {

const numberArr = []; // Array to store elements and their indices

for (let i = 0; i < nums.length; i++) {

const diff = target - nums[i];

for (let j = 0; j < numberArr.length; j++) {

if (numberArr[j][0] === diff) {

return [numberArr[j][1], i]; /

}

}

numberArr.push([nums[i], i]);

}

return [];

}

const nums = [2, 7, 8, 15];

const target = 10;

const result = twoSum(nums, target);

console.log(result); // Output: [0, 1]

**2nd Solution**

function removeElement(nums, val) {

let i = 0; // Pointer for iterating through the array

let j = 0; // Pointer for placing elements not equal to val

while (i < nums.length) {

if (nums[i] !== val) {

nums[j] = nums[i];

j++;

}

i++;

}

return j;

}

let nums = [3,2,2,3,5]

let val = 5

console.log(removeElement(nums, val))

**3rd Solution**

function searchInsert(nums, target) {

let left = 0;

let right = nums.length - 1;

while (left <= right) {

let mid = Math.floor((left + right) / 2);

if (nums[mid] === target) {

return mid;

} else if (nums[mid] < target) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return left;

}

const nums = [1, 2, 10, 7];

const target = 4;

const result = searchInsert(nums, target);

console.log(result);

**4th Solution**

function plusOne(digits) {

const n = digits.length;

let carry = 1;

for (let i = n - 1; i >= 0; i--) {

digits[i] += carry;

carry = Math.floor(digits[i] / 10);

digits[i] %= 10;

if (carry === 0) {

break;

}

}

if (carry === 1) {

digits.unshift(carry);

}

return digits;

}

const digits = [1, 2, 5];

const result = plusOne(digits);

console.log(result); // Output: [1, 2, 6]

**6th Solution**

function containsDuplicate(nums) {

const occcurence = [];

for (const num of nums) {

if (occcurence.includes(num)) {

return true;

}

occcurence.push(num);

}

return false;

}

const nums = [1,2,3,1];

const result = containsDuplicate(nums);

console.log(result);

**7th Solution**

function moveZeroes(nums) {

let left = 0;

let right = 0;

while (right < nums.length) {

if (nums[right] !== 0) {

[nums[left], nums[right]] = [nums[right], nums[left]];

left++;

}

right++;

}

}

let nums = [0, 1, 0, 3, 12];

moveZeroes(nums);

console.log(nums);

**8th Solution**

function findErrorNums(nums) {

const setNums = new Set();

let duplicate = -1;

for (const num of nums) {

if (setNums.has(num)) {

duplicate = num;

} else {

setNums.add(num);

}

}

let missing = -1;

for (let i = 1; i <= nums.length; i++) {

if (!setNums.has(i)) {

missing = i;

break;

}

}

return [duplicate, missing];

}

let nums = [1,2,2,4];

console.log(findErrorNums(nums));