Naga Rajasekhar

Arrays - 2nd Assignment

**1st Solution**

function arrayPairSum(nums) {

nums.sort((a, b) => a - b);

let result = 0;

for (let i = 0; i < nums.length; i += 2) {

result += nums[i];

}

return result;

}

nums=[1,4,3,2]

console.log(arrayPairSum(nums))

**2nd Solution**

function maxCandies(candyType) {

const maxTypes = candyType.length / 2;

const uniqueTypes = [];

for (const candy of candyType) {

if (!uniqueTypes.includes(candy)) {

uniqueTypes.push(candy);

}

if (uniqueTypes.length >= maxTypes) {

break;

}

}

return uniqueTypes.length;

}

candyType = [1, 1, 2, 2, 3, 3, 4, 4]

console.log(maxCandies(candyType))

**3rd Solution**

function findLHS(nums) {

const count = new Array(2 \* Math.pow(10, 4) + 1).fill(0);

let maxLength = 0;

// Count the frequencies of each number

for (const num of nums) {

count[num] += 1;

}

// Iterate through the count array

for (let i = 0; i < count.length - 1; i++) {

if (count[i] > 0 && count[i + 1] > 0) {

const length = count[i] + count[i + 1];

maxLength = Math.max(maxLength, length);

}

}

return maxLength;

}

**4th Solution**

function canPlaceFlowers(flowerbed, n) {

const length = flowerbed.length;

let count = 0;

let i = 0;

while (i < length) {

if (

flowerbed[i] === 0 &&

(i === 0 || flowerbed[i - 1] === 0) &&

(i === length - 1 || flowerbed[i + 1] === 0)

) {

flowerbed[i] = 1;

count++;

i += 2;

} else {

i++;

}

}

return count >= n;

}

console.log(canPlaceFlowers([1,0,0,0,1], 1))

**5th Solution**

function maximumProduct(nums) {

nums.sort((a, b) => a - b);

const n = nums.length;

const maxProduct = nums[n - 1] \* nums[n - 2] \* nums[n - 3];

const possibleMaxProduct = nums[0] \* nums[1] \* nums[n - 1];

return Math.max(maxProduct, possibleMaxProduct);

}

console.log((maximumProduct([1,2,3])))

**6th Solution**

function search(nums, target) {

let left = 0;

let right = nums.length - 1;

while (left <= right) {

const 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 -1;

}

console.log(search([-1,0,3,5,9,12], 9))

**7th Solution**

function isMonotonic(nums) {

let increasing = true;

let decreasing = true;

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

if (nums[i] < nums[i - 1]) {

increasing = false;

}

if (nums[i] > nums[i - 1]) {

decreasing = false;

}

}

return increasing || decreasing;

}

console.log(isMonotonic([1,2,2,3]))