Naga Rajasekhar

Arrays - 3rd Assignment

**1st Solution**

function threeSumClosest(nums, target) {

const n = nums.length;

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

let closestSum = Infinity;

for (let i = 0; i < n - 2; i++) {

let left = i + 1;

let right = n - 1;

while (left < right) {

const sum = nums[i] + nums[left] + nums[right];

if (Math.abs(sum - target) < Math.abs(closestSum - target)) {

closestSum = sum;

}

if (sum === target) {

return sum;

} else if (sum < target) {

left++;

} else {

right--;

}

}

}

return closestSum;

}

console.log(threeSumClosest([-1,2,1,-4], 1))

**2nd Solution**

function fourSum(nums, target) {

const n = nums.length;

const result = [];

if (n < 4) {

return result;

}

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

for (let i = 0; i < n - 3; i++) {

if (i > 0 && nums[i] === nums[i - 1]) {

continue;

}

for (let j = i + 1; j < n - 2; j++) {

if (j > i + 1 && nums[j] === nums[j - 1]) {

continue;

}

let left = j + 1;

let right = n - 1;

while (left < right) {

const sum = nums[i] + nums[j] + nums[left] + nums[right];

if (sum === target) {

result.push([nums[i], nums[j], nums[left], nums[right]]);

while (left < right && nums[left] === nums[left + 1]) {

left++;

}

while (left < right && nums[right] === nums[right - 1]) {

right--;

}

left++;

right--;

} else if (sum < target) {

left++;

} else {

right--;

}

}

}

}

return result;

}

console.log(fourSum([1,0,-1,0,-2,2], 0))

**4th Solution**

function searchInsert(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 left;

}

console.log(searchInsert([1,3,5,6], 5))

**5th Solution**

function plusOne(digits) {

const n = digits.length;

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

digits[i]++;

if (digits[i] === 10) {

digits[i] = 0;

} else {

return digits;

}

}

digits.unshift(1);

return digits;

}

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

**6th Solution**

function singleNumber(nums) {

let result = 0;

for (let num of nums) {

result ^= num;

}

return result;

}

console.log(singleNumber([2,2,1, 1, 4]))

**7th Solution**

function findMissingRanges(nums, lower, upper) {

const result = [];

let prev = lower - 1;

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

let curr = (i === nums.length) ? upper + 1 : nums[i];

if (curr - prev > 1) {

result.push(formatRange(prev + 1, curr - 1));

}

prev = curr;

}

return result;

}

function formatRange(start, end) {

if (start === end) {

return start.toString();

} else {

return `${start}-${end}`;

}

}

console.log(findMissingRanges([0,1,3,50,75], 0, 99))

**8th Solution**

function canAttendMeetings(intervals) {

intervals.sort((a, b) => a[0] - b[0]);

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

if (intervals[i][0] < intervals[i - 1][1]) {

return false;

}

}

return true;

}

const intervals = [[0, 30], [5, 10], [15, 20]];

console.log(canAttendMeetings(intervals));