Practice Problems

Relevel
by Unacademy

Problem - 1

Given an array of length N, where every element is present twice and only one element is present once. Find the unique element

Sample input: N = 5, Array = [3,2,1,2,3]
Sample output: 1

Problem - 1

Solution: https://jsfiddle.net/vuny9ftw/

```
function findSingle(ar, N) {
// Do XOR of all elements and return
let res = ar[0];
for (let i = 1; i < N; i++)
res = res ^ ar[i];
return res;
}
let Array = [3,2,1,2,3];
let N = 5;
document.write("Element occurring once is " +
findSingle(Array, N));
```

Problem - 2

Find the pair of elements in a given array A such that the sum of the pair is equal to N

Input: A = [10, 20, 10, 40, 50 , 70], N=50
Output: 2,3

Problem - 2

Solution: https://jsfiddle.net/ur3cabj1/

```
function sumOfPair(nums, target_num) {
var map = [];
var indexes = [];
for (var idx = 0; idx < nums.length; idx++){
if (map[nums[idx]] != null){
var index = map[nums[idx]];
indexes[0] = index;
indexes[1] = idx;
break;}
else{
map[target_num - nums[idx]] = idx; } }
return indexes; }
console.log(sumOfPair([10,20,10,40,50,60,70],50));
```

Problem - 3

A subarray of an array is defined as the contiguous cross section of the array. Example: [1,2,3] has the following subarrays: [1],[2],[3],[1,2],[2,3].[1,2,3] Given an array print all the subarrays of the given array

Example-1

Input: [1,2,3]

Output: [1],[2],[3],[1,2],[2,3].[1,2,3]

Problem - 3

Solution: https://jsfiddle.net/j48nuyxw/

```
function subArray(n) {
for (let i = 0; i < n; i++) {
for (let j = i; j < n; j++) {
for (let k = i; k <= j; k++)
document.write(arr[k] + " ");
document.write("</br>");
}
}
}
let arr = [1, 2, 3];
subArray(arr.length);
```

Problem - 4

You have an array of n elements. Your job is to find the element that is in majority.
Any element whose count is greater than n/2 will be considered as a majority element.

Example-1:
Input: [3,1,3,3,2]
Output: 3

Problem - 4

Solution: https://jsfiddle.net/gb6rp930/

```
function findMajority(arr, n)
{
let maxCount = 0;
let index = -1;
for(let i = 0; i < n; i++)
{
let count = 0;
for(let j = 0; j < n; j++)
{
if (arr[i] == arr[j])
count++;
}
```

Problem - 4

```
if (count > maxCount)
{
maxCount = count;
index = i;
}
}
if (maxCount > n / 2)
document.write(arr[index]);
else
document.write("No Majority Element");
}
let arr = [3,1,3,3,2];
let n = arr.length;
findMajority(arr, n);
```

Problem - 5

Find the intersection of two sorted arrays. OR in other words, Given 2 sorted arrays, find all the elements which occur in both the arrays.

Example-1

Input: A: [1 2 3 3 4 5 6] B: [3 3 5]

Output: [3 3 5]

Problem - 5

Solution: https://jsfiddle.net/gx8rt29z/

```
function printIntersection(arr1, arr2, m, n) {
var i = 0,
j = 0;
while (i < m && j < n) {
if (arr1[i] < arr2[j])
i++;
else if (arr2[j] < arr1[i])
j++;
```

Problem - 5

```
else {
document.write(arr2[j++] + " ");
i++;
}
}
}
var arr1 = [1, 2, 3, 3, 4, 5, 6];
var arr2 = [3, 3, 5];
var m = arr1.length;
var n = arr2.length;
printIntersection(arr1, arr2, m, n);
```

Problem - 6

Given an array . Your task is to find if there is a triplet present with the given sum.

Input - [1,2,5,6,7,8,3], sum = 8
Output - 1,2,5

Problem - 6

Solution: https://jsfiddle.net/jhxrv58p/

```
function findTriplet(A, arr_size, sum)
{
  let l, r;

  A.sort((a,b) => a-b);
  for (let i = 0; i < arr_size - 2; i++) {


    l = i + 1;

    r = arr_size - 1;
    while (l < r) {
      if (A[i] + A[l] + A[r] == sum)
      {
      console.log("Triplet is " + A[i] + ", "
        + A[l] + ", " + A[r]);
        return true;
      }
      else if (A[i] + A[l] + A[r] < sum)
        l++;
      else
        r--;
    }
  }

  return false;
}



  let A = [1,2,5,6,7,8,3 ];
  let sum = 8;
  let arr_size = A.length;

  findTriplet(A, arr_size, sum);
```

Problem - 7

Given a 2D array. Our task is to find the unique elements. A unique element is an element whose frequency is 1 i.e. it is not repeating in the whole 2D Array.

If there is no any unique element, print message as “No unique element found”

Input:

2	14	15	18
10	18	14	22
8	21	22	15
10	14	21	28

Output: 2, 8, 28

Problem - 7

Code Link -> https://jsfiddle.net/j38uhwve/

JavaScript + No-Library (pure JS) ▾

```
var R = 4, C = 4;

// Function that calculate
// unique element
function unique(mat, n, m)
{
  var maximum = 0, flag = 0;
  for(var i = 0; i < n; i++)
    for(var j = 0; j < m; j++)

      // Find maximum element in a matrix
      if (maximum < mat[i][j])
        maximum = mat[i][j];

  // Take 1-D array of (maximum + 1) size
  var b = Array(maximum+1).fill(0);
  for(var i = 0 ; i < n; i++)
    for(var j = 0; j < m; j++)
      b[mat[i][j]]++;

  // Print unique element
  for(var i = 1; i <= maximum; i++)
    if (b[i] == 1)
      console.log(i + " ");
      flag = 1;

  if (flag == 0)
  {
    console.log("No unique element found");
  }
}

//Input array
var mat = [ [ 11, 2, 2, 20 ],
      [ 5, 16, 20, 7 ],
      [ 1, 13, 5, 16 ],
      [ 6, 7, 18, 15 ] ];

//execute the logic
unique(mat, R, C);
```

Problem - 8

Given a grid of size NxM, K cells have plants while remaining have weeds. Your job is to lay fence on the grid such that the following conditions are met: If adjacent cells are plants no need to put fence between them If adjacent cell is of weed, then put a fence

Note: Cells are said to be adjacent if they share the same boundary.
And building of fences can happen on cell boundaries. So now the total length of constructed fence will be calculated as count of pairs of the side adjacent cells such that there is a fence built on there common side on sides of cells. On the grid boundary which have fences built on them. Find the minimum required length of the total length of fences that needs to be built.

Problem - 8

Example-1

Input:

n = 4;

m = 4;

k = 1;

arr = [[1,1]];

Output:

4

Problem - 8

Solution: https://jsfiddle.net/u2fmtg8L/

```
var N, M, K;
N = 4;
M = 4;
K = 1;
var items = [
  [1, 1]
];
var set = [];
for ( i = 0; i < K; i++) {

   set[i] = items[i][0] + " " + items[i][1];


}
var value = K * 4;
console.log(set);
for (i = 0; i < K; i++) {
  s1 = items[i][0] + " " + (items[i][1] - 1);
  s2 = items[i][0] + " " + (items[i][1] + 1);
  s3 = (items[i][0] - 1) + " " + items[i][1];
  s4 = (items[i][0] + 1) + " " + items[i][1];
  console.log(s1+" "+s2+" "+s3+" "+s4);
  if (set.includes(s1))
    value--;
  if (set.includes(s2))
    value--;
  if (set.includes(s3))
    value--;
  if (set.includes(s4))
    value--;
}
console.log(value);
```

Thank You!