JavaScript Problem Set 5 (PS5) Solutions

Problem 1 Find the Pair (Two Sum Revisited)

Write a function `hasPairWithSum(arr, target)` that returns true if any two numbers in the array add up to the target.

Solution:

```
function hasPairWithSum(arr, target) {
let set = new Set();
for (let num of arr) {
if (set.has(target - num)) return true;
set.add(num);
}
return false;
}
```

Problem 2 Max Sum Subarray of Size K

Write `maxSubarraySum(arr, k)` that returns the maximum sum of a contiguous subarray of size k.

Solution:

```
function maxSubarraySum(arr, k) {
let max = 0, temp = 0;
for (let i = 0; i < k; i++) temp += arr[i];
max = temp;
for (let i = k; i < arr.length; i++) {
temp = temp - arr[i - k] + arr[i];
max = Math.max(max, temp);
}
return max;
}
```

Problem 3 Anagram Checker

Write a function `isAnagram(str1, str2)` that checks if two strings are anagrams.

Solution:

```
function isAnagram(str1, str2) {
return str1.split('').sort().join('') === str2.split('').sort().join
   ('');
}
```

Problem 4 Move Zeroes to End

Write `moveZeroes(arr)` that moves all 0s to the end while keeping order.

Solution:

```
function moveZeroes(arr) {
let insertPos = 0;
for (let num of arr) {
if (num !== 0) arr[insertPos++] = num;
}
while (insertPos < arr.length) arr[insertPos++] = 0;
return arr;
}
```

Problem 5 Merge Two Sorted Arrays

Write `mergeSorted(arr1, arr2)` that merges and sorts two arrays.

Solution:

```
function mergeSorted(arr1, arr2) {
let i = 0, j = 0, result = [];
while (i < arr1.length && j < arr2.length) {
if (arr1[i] < arr2[j]) result.push(arr1[i++]);
else result.push(arr2[j++]);
}
return result.concat(arr1.slice(i)).concat(arr2.slice(j));
}
```

Problem 6 Promise Timeout

Create `wait(ms)` that resolves after `ms` milliseconds.

Solution:

```
function wait(ms) {
return new Promise(resolve => setTimeout(resolve, ms));
}
```

Problem 7 Count Vowels

Write countVowels(str) that counts vowels in a string.

Solution:

```
function countVowels(str) {
return (str.match(/[aeiou]/gi) || []).length;
}
```

Problem 8 Product of Array Except Self

Write productExceptSelf(arr) to return a new array with the product of all other elements.

Solution:

```
function productExceptSelf(arr) {
let n = arr.length;
let left = Array(n).fill(1);
let right = Array(n).fill(1);
let result = Array(n);

for (let i = 1; i < n; i++) left[i] = left[i-1] * arr[i-1];
for (let i = n - 2; i >= 0; i--) right[i] = right[i+1] * arr[i+1];
for (let i = 0; i < n; i++) result[i] = left[i] * right[i];

return result;
}
```

Problem 9 Find Missing Number

Given [1, 2, 4] return missing number from 1 to n.

Solution:

```
function findMissing(arr) {
let n = arr.length + 1;
let total = (n * (n + 1)) / 2;
return total - arr.reduce((a, b) => a + b, 0);
}
```

Problem 10 Async Fetch Mock

Write an async function getUser() that simulates an API fetch.

Solution:

```
async function getUser() {
await new Promise(resolve => setTimeout(resolve, 1000));
return "User fetched!";
}
```