JavaScript Problem Set 4 (PS4) Solutions

Problem 1 Reverse Words in a Sentence

Write a function `reverseWords(sentence)` that reverses the order of words.

Example:

```
reverseWords("I love JavaScript"); //     "JavaScript love I"
```

Solution:

```
function reverseWords(sentence) {
  return sentence.split(" ").reverse().join(" ");
}
```

Problem 2 Character Frequency Counter

Write a function `charFrequency(str)` that counts character frequency (ignore spaces).

Example:

```
charFrequency("hello world");
//     { h:1, e:1, l:3, o:2, w:1, r:1, d:1 }
```

Solution:

```
function charFrequency(str) {
  let freq = {};
  for (let char of str.replace(/\s/g, '')) {
    freq[char] = (freq[char] || 0) + 1;
  }
  return freq;
}

// This is a simpler way (or easier to understand):

function charFrequency(str) {
  let freq = {};
  for (let char of str.replace(/\s/g, '')) {
    if (freq[char]) {                // if it already exists
      freq[char] = freq[char] + 1;
    } else {
```

```
      freq[char] = 1;
    }
  }
  return freq;
}
```

Problem 3 FizzBuzz

Write `fizzBuzz(n)` that prints 1 to `n`, with:

- "Fizz" for multiples of 3
- "Buzz" for multiples of 5
- "FizzBuzz" for both

Solution:

```
function fizzBuzz(n) {
  for (let i = 1; i <= n; i++) {
    if (i % 15 === 0) console.log("FizzBuzz");
    else if (i % 3 === 0) console.log("Fizz");
    else if (i % 5 === 0) console.log("Buzz");
    else console.log(i);
  }
}
```

Problem 4 Palindrome Checker

Write `isPalindrome(str)` to check if a string is a palindrome (ignore spaces/case).

Examples:

```
isPalindrome("racecar"); // true
isPalindrome("nurses run"); // true
```

Solution:

```
function isPalindrome(str) {
  let clean = str.replace(/\s+/g, '').toLowerCase();
  return clean === clean.split('').reverse().join('');
}
```

Problem 5 Remove Duplicates from Array

Write `removeDuplicates(arr)` to return a new array with no duplicates.

Example:

```
removeDuplicates([1, 2, 2, 3]); //     [1, 2, 3]
```

Solution:

```
function removeDuplicates(arr) {
  return [...new Set(arr)];
}
```

Problem 6 Flatten Nested Array

Write `flatten(arr)` that flattens nested arrays (no `.flat()`!).

Example:

```
flatten([1, [2, [3, 4]], 5]); //     [1, 2, 3, 4, 5]
```

Solution:

```
function flatten(arr) {
  let result = [];
  for (let el of arr) {
    if (Array.isArray(el)) {
      result.push(...flatten(el));
    } else {
      result.push(el);
    }
  }
  return result;
}

\\ Also, this one:

function flatten(arr) {
  let result = [];
  for (let el of arr) {
    if (el instanceof Array) {   // check if it's an array
      result.push(...flatten(el));
    } else {
      result.push(el);
    }
  }
  return result;
}
```

Problem 7 Longest Word in Sentence

Write `longestWord(sentence)` that returns the longest word.

Example:

```
longestWord("The quick brown fox"); //     "quick"
```

Solution:

```
// The simplest way is probably this one:

function longestWord(sentence) {
  let array_1 = sentence.split(/[^a-zA-Z]/).filter(Boolean); //
     filters by any delimiter that isn't an alphabetic character
  let longest = "";
  for (let word of array_1) {
    if (word.length > longest.length) {
      longest = word;
    }
  }
  return longest;
}

// Another way is:

function longestWord(sentence) {
  return sentence.split(" ").reduce((longest, word) =>
    word.length > longest.length ? word : longest
  , "");
}
```

Problem 8 First Non-Repeating Character

Write `firstUniqueChar(str)` that returns the first non-repeating character.

Example:

```
firstUniqueChar("aabbcdd"); //     "c"
```

Solution:

```
function firstUniqueChar(str) {
  let freq = {};
  for (let char of str) {
    freq[char] = (freq[char] || 0) + 1;
  }
  for (let char of str) {
    if (freq[char] === 1) return char;
  }
  return null;
```

```
}

// Again, probably easier to read this one:

function firstUniqueChar(str) {
  let freq = {};
  for (let char of str) {
    if (freq[char]) {
      freq[char] += 1;
    } else {
      freq[char] = 1;
    }
  }
  for (let char of str) {
    if (freq[char] === 1) {
      return char;
    }
  }
  return null;
}
```

Problem 9 Factorial (Recursion)

Write the factorial function `factorial(n)` using recursion and another method.

Example:

```
factorial(5); //     120
```

Solution:

```
function factorial(n) {
  if (n <= 1) return 1;
  return n * factorial(n - 1);
}

// I prefer this next solution, I guess it underscores the nature of
    the factorial better.

function factorial(n) {
  let p = 1;
  for (let i = 1; i <= n; i++) {
    p *= i;
  }
  return p;
}
```

Problem 10 Two Sum

Write `twoSum(nums, target)` that returns indices of two numbers adding to target.

Example:

```
twoSum([2, 7, 11, 15], 9); // [0, 1]
```

Solution:

```
function twoSum(nums, target) {
  let map = {};
  for (let i = 0; i < nums.length; i++) {
    let diff = target - nums[i];
    if (map[diff] !== undefined) {
      return [map[diff], i];
    }
    map[nums[i]] = i;
  }
}

// I repeated this one in the next problem set.
```