

Given an array of numbers find its moving average with window length 5. Try to do this using only "2 * n" number of additions where n is the length of the original array.

Example Input: [1, 2, 3, 4, 5, 6, 7, 8]

Example Output: [3, 4, 5, 6]

```
function movingAvg(arr) {
  const movingAves = [];
  let sum = 0;

  // Calculate sum of first window
  for (let i = 0; i < 5; i++) {
    sum += arr[i];
  }
  movingAves.push(sum / 5);

  // Update sum for subsequent windows using previous sum
  for (let i = 1; i <= arr.length - 5; i++) {
    sum = sum - arr[i - 1] + arr[i + 4]; // Subtract first element of previous window, add
    next element
    movingAves.push(sum / 5);
  }

  return movingAves;
}

console.log(movingAvg([1, 2, 3, 4, 5, 6, 7, 8]));
```

Without the 2 * n condition:

```
function movingAvg(arr) {
  const movingAves = [];

  for (let i = 0; i <= arr.length - 5; i++) {
    let sum = 0;
    for (let j = i; j < i + 5; j++) {
      sum += arr[j];
    }
    movingAves.push(sum / 5);
  }
  return movingAves;
}

console.log(movingAvg([1, 2, 3, 4, 5, 6, 7, 8]));
```