

Test Suite 1

passes: 6 failures: 0 duration: 0.03s

100%

primefunctions

primeGen()

- ✓ primeGen(10) = [2, 3, 5, 7]
- ✓ primeGen(20) = [2, 3, 5, 7, 11, 13, 17, 19]

cumulativeSum()

- ✓ cumulativeSum([1, 2, 3, 4]) = [1, 3, 6, 10]
- ✓ cumulativeSum([10, 11, 12, 13, 14]) = [10, 21, 33, 46, 60]

maxPrimeSum()

- ✓ maxPrimeSum(100) = [41, 6]
- ✓ maxPrimeSum(1000) = [953, 21]

Test Suite 2

Original code from HW4

passes: 2 failures: 0 duration: 6.28s

100%

primefunctionsOriginal

maxPrimeSum()

- ✓ maxPrimeSum(10000) should take less than 75ms 72ms

```
this.timeout(85);  
this.slow(0);  
maxPrimeSum(10000);
```

- ✓ maxPrimeSum(100000) should take less than 3000ms 6184ms

```
this.timeout(7500);  
this.slow(0);  
maxPrimeSum(100000);
```

Optimized 1 code from HW5

passes: 2 failures: 0 duration: 4.26s 100%

primefunctionsOptimized

maxPrimeSum()

✓ maxPrimeSum(10000) should take less than 75ms 64ms

```
this.timeout(85);  
this.slow(0);  
maxPrimeSum(10000);
```

✓ maxPrimeSum(100000) should take less than 3000ms 4171ms

```
this.timeout(7500);  
this.slow(0);  
maxPrimeSum(100000);
```

Optimized 2 code from HW5

passes: 2 failures: 0 duration: 4.17s 100%

primefunctionsOptimized2

maxPrimeSum()

✓ maxPrimeSum(10000) should take less than 75ms 65ms

```
this.timeout(75);  
this.slow(0);  
maxPrimeSum(10000);
```

✓ maxPrimeSum(100000) should take less than 3000ms 4082ms

```
this.timeout(8000);  
this.slow(0);  
maxPrimeSum(100000);
```