Week 02

Seth Childers

1 Exercise 01

Problem: For each of the following algorithms, indicate (i) a natural size metric for its inputs, (ii) its basic operation, and (iii) whether the basic operation count can be different for inputs of the same size

Code written in Markdown

```
# Exercises 2.1 - #01
## For each of the following algorithms, indicate
 * (i) a natural size metric for its inputs,
 * (ii) its basic operation, and
 * (iii) whether the basic operation count can be different for inputs of the
    same size:

| Algorithm | Natural Size Metric of Input | Basic Operation | Basic Operation
   Count Can Be Different than Input Size |
| :--------------: | :---: | :--------------------------------: | :---: |
| a - computing the sum of *n* numbers | n | addition of two numbers | no |
| b - computing *n*! | n | multiplication of two numbers | no |
| c - finding the largest element in a list of *n* numbers | n | comparing the
   value of two elements | yes |
| d - Euclid's algorithm | n | Modulus | no |
| e - sieve of Eratosthenes | n | multiplication of two numbers | no |
| f - pen-and-pencil algorithm for multiplying two *n*-digit decimal integers
   | n | multiplication of two numbers | no |
```

Results

Exercises 2.1 - #01

For each of the following algorithms, indicate

- (i) a natural size metric for its inputs,
- (ii) its basic operation, and
- (iii) whether the basic operation count can be different for inputs of the same size:

Algorithm	Natural Size Metric of Input	Basic Operation	Basic Operation Count Can Be Different than Input Size
a - computing the sum of *n* numbers	n	addition of two numbers	no
b - computing *n*!	n	multiplication of two numbers	no
c - finding the largest element in a list of *n* numbers	n	comparing the value of two elements	yes
d - Euclid's algorithm	n	Modulus	no
e - sieve of Eratosthenes	n	multiplication of two numbers	no
f - pen-and-pencil algorithm for multiplying two *n*-digit decimal integers	n	multiplication of two numbers	no

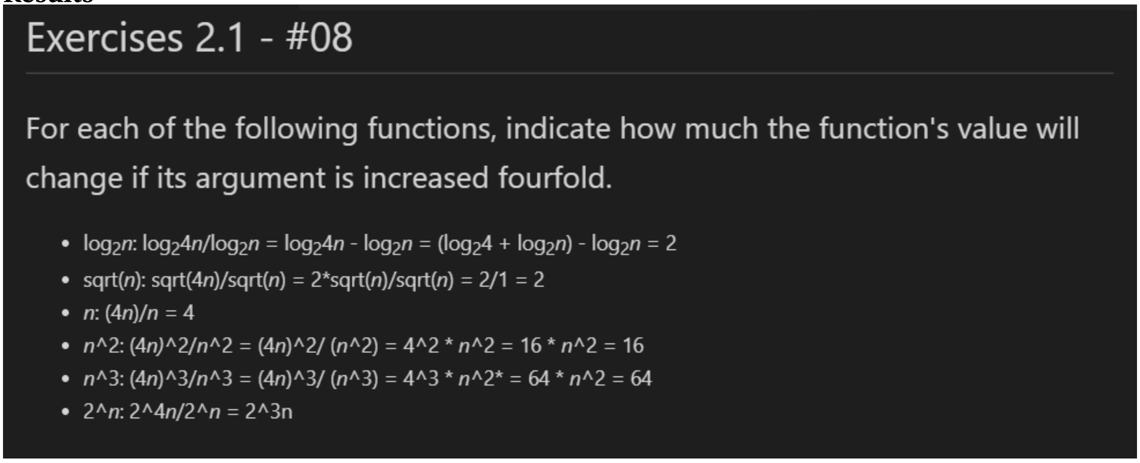
2 Exercise 02

Problem: For each of the following functions, indicate how much the function's value will change if its argument is increased fourfold.

Code written in Markdown

```
# Exercises 2.1 - #08
## For each of the following functions, indicate how much the function‘s value will change if its argument is increased fourfold.
 * log<sub>2</sub>*n*: log<sub>2</sub>4*n*/log<sub>2</sub>*n* = log<sub>2</sub>4*n* - log<sub>2</sub>*n* = (log<sub>2</sub>4 + log<sub>2</sub>*n*) - log<sub>2</sub>*n* = 2
 * sqrt(*n*): sqrt(4*n*)/sqrt(*n*) = 2*sqrt(*n*)/sqrt(*n*) = 2/1 = 2
 * *n*: (4*n*)/*n* = 4
 * *n*^2: (4*n)*^2/*n*^2 = (4*n*)^2/ (*n*^2) = 4^2 * *n*^2 = 16 * *n*^2 = 16
 * *n*^3: (4*n*)^3/*n*^3 = (4*n*)^3/ (*n*^3) = 4^3 * *n*^2* = 64 * *n*^2 = 64
 * 2^*n*: 2^4*n*/2^*n* = 2^3n
```

Results



Exercises 2.1 - #08

For each of the following functions, indicate how much the function's value will change if its argument is increased fourfold.

- $\log_2 n$: $\log_2 4n / \log_2 n$ = $\log_2 4n$ - $\log_2 n$ = ($\log_2 4$ + $\log_2 n$) - $\log_2 n$ = 2
- sqrt(*n*): sqrt(4*n*)/sqrt(*n*) = 2*sqrt(*n*)/sqrt(*n*) = 2/1 = 2
- *n*: (4*n*)/*n* = 4
- *n*^2: (4*n*)^2/*n*^2 = (4*n*)^2/ (*n*^2) = 4^2 * *n*^2 = 16 * *n*^2 = 16
- *n*^3: (4*n*)^3/*n*^3 = (4*n*)^3/ (*n*^3) = 4^3 * *n*^2* = 64 * *n*^2 = 64
- 2^*n*: 2^4*n*/2^*n* = 2^3n

3 Exercise 03

Problem: For each of the following functions, indicate how much the function's value will change if its argument is increased fourfold.

Code written in Markdown

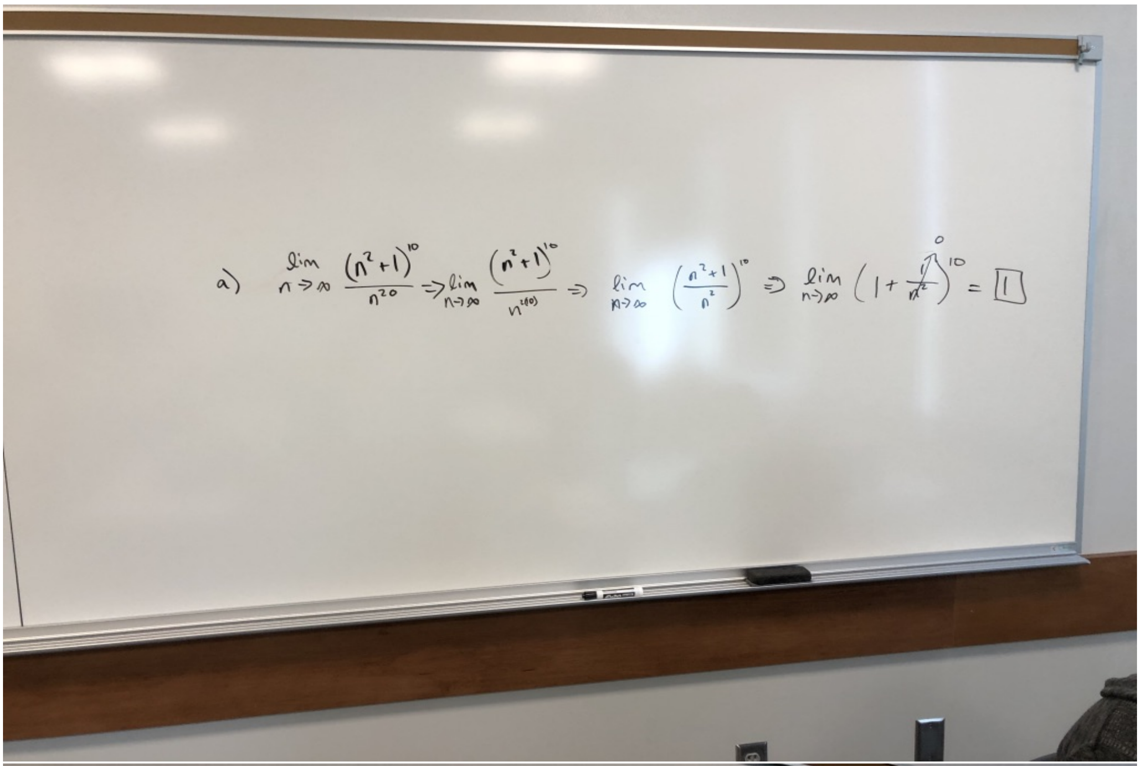
```
# Exercise 2.2 - #03
## For each of the following functions, indicate the class ÎŸ(g(*n*)) the
    function belongs to. (Use the simplest g(*n*) possible in your answers.)
    Prove your assertion.

|Function| Code |
| :----: | :---: |
| a | (n<sup>2</sup> + 1)<sup>10</sup> |
| b | sqrt(10*n*<sup>2</sup> + 7*n* + 3) |
| c | 2*n* lg(*n* + 2)<sup>2</sup> + (*n* + 2)<sup>2</sup> lg n/2 |
| d | 2<sup>*n* + 1</sup> + 3<sup>*n* - 1</sup> |
| e | âŇŁ log<sub>2</sub>*n* âŇŃ |
```

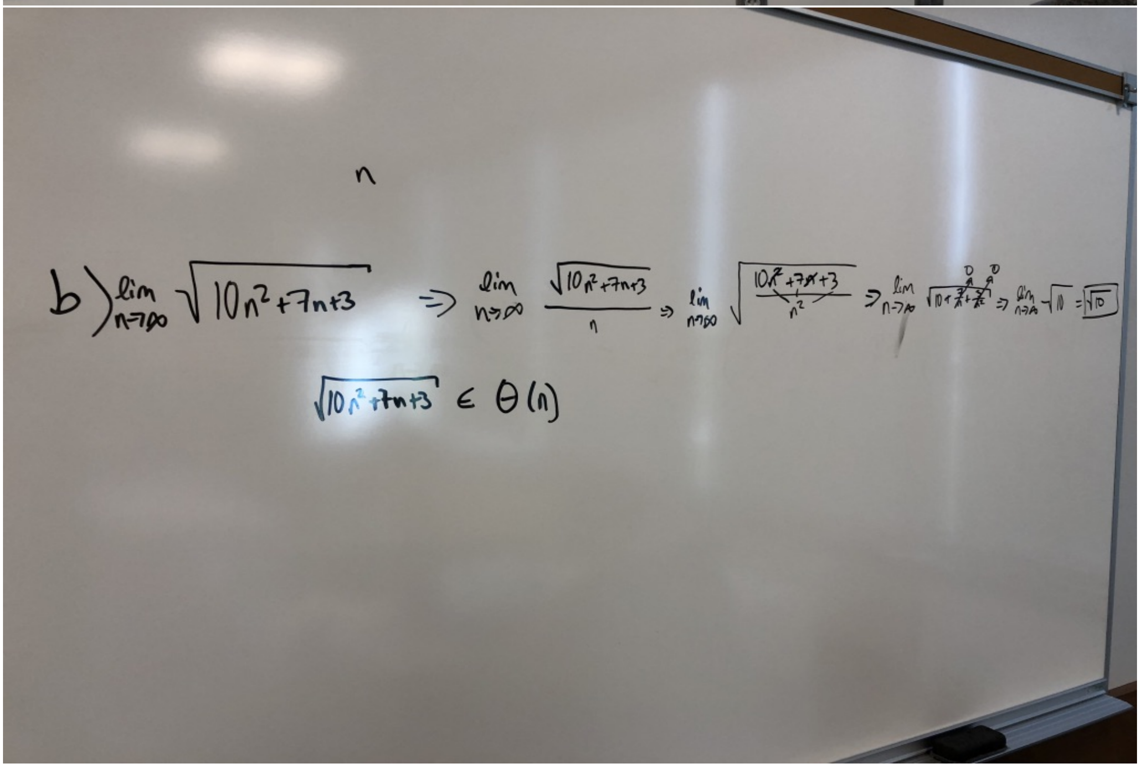
Results

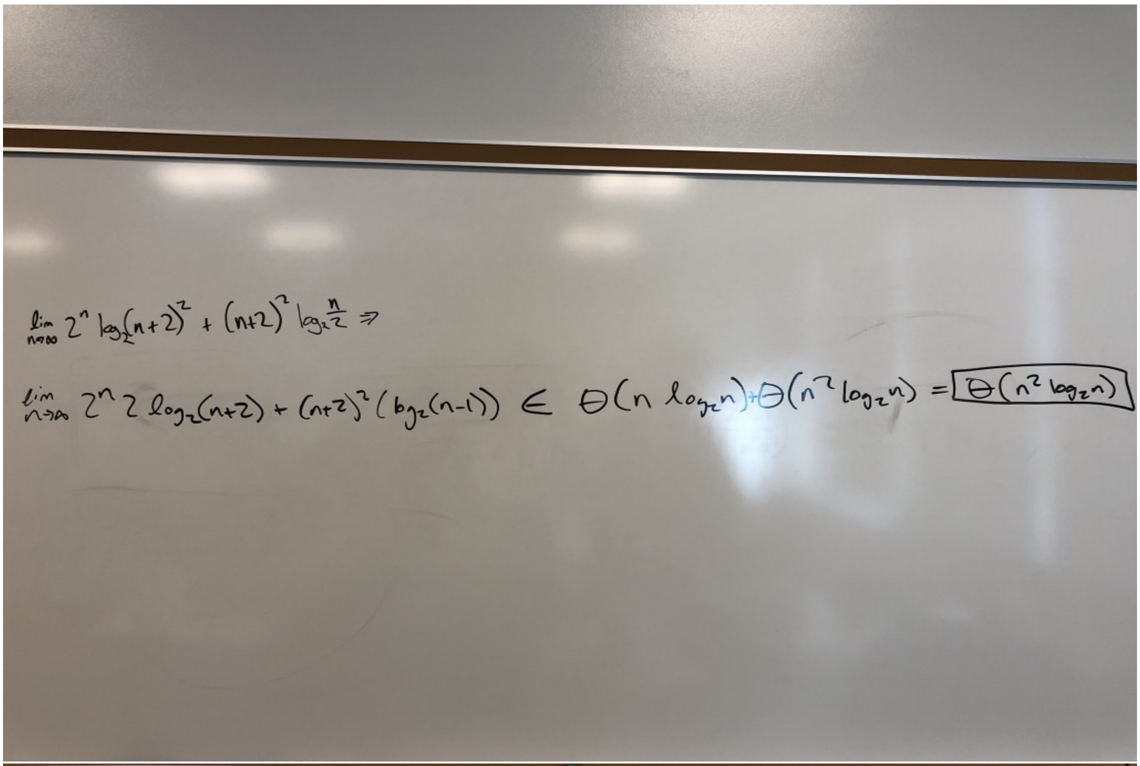
Exercise 2.2 - #03

For each of the following functions, indicate the class Θ(g(*n*)) the function belongs to. (Use the simplest g(*n*) possible in your answers.) Prove your assertion.

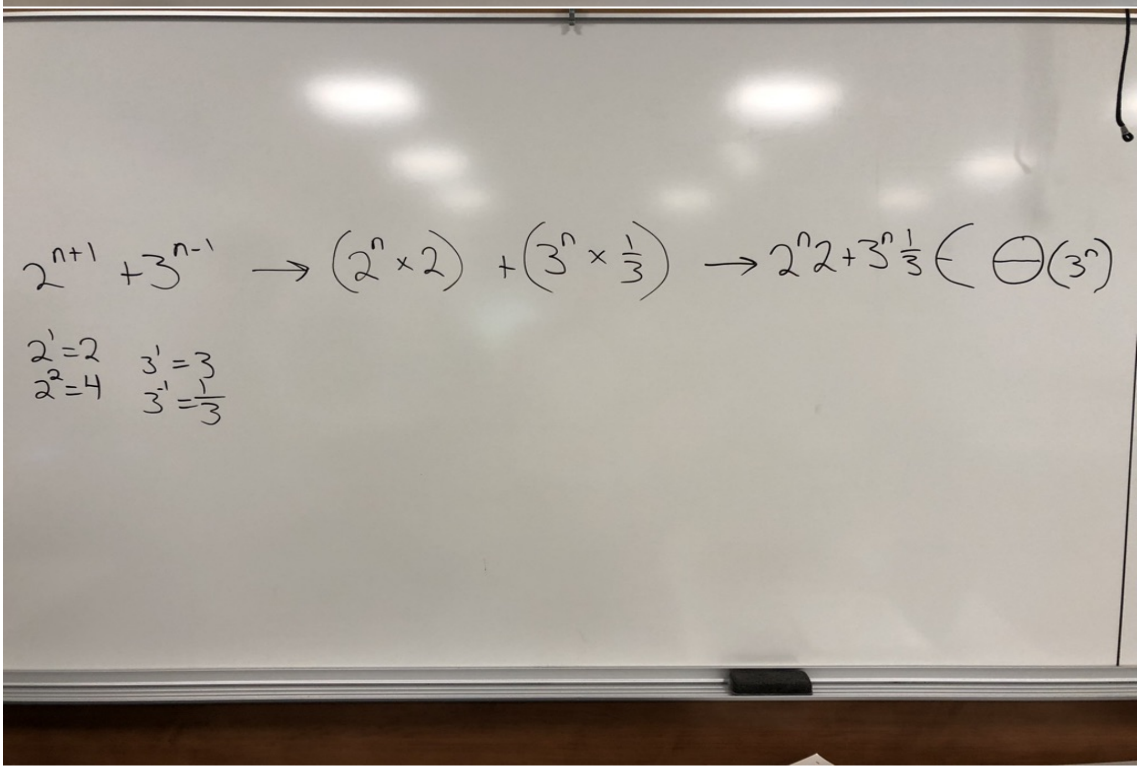
Function	Code
a	$(n^2 + 1)^{10}$
b	sqrt($10n^2 + 7n + 3$)
c	$2n \lg(n + 2)^2 + (n + 2)^2 \lg n/2$
d	$2^{n + 1} + 3^{n - 1}$
e	$\lfloor \log_2 n \rfloor$


a) $\lim_{n\to\infty} \frac{(n^2+1)^{10}}{n^{20}} \Rightarrow \lim_{n\to\infty} \frac{(n^2+1)^{10}}{n^{2(10)}} \Rightarrow \lim_{n\to\infty} \left(\frac{n^2+1}{n^2}\right)^{10} \Rightarrow \lim_{n\to\infty} \left(1+\frac{1}{n^2}\right)^{10} = \boxed{1}$

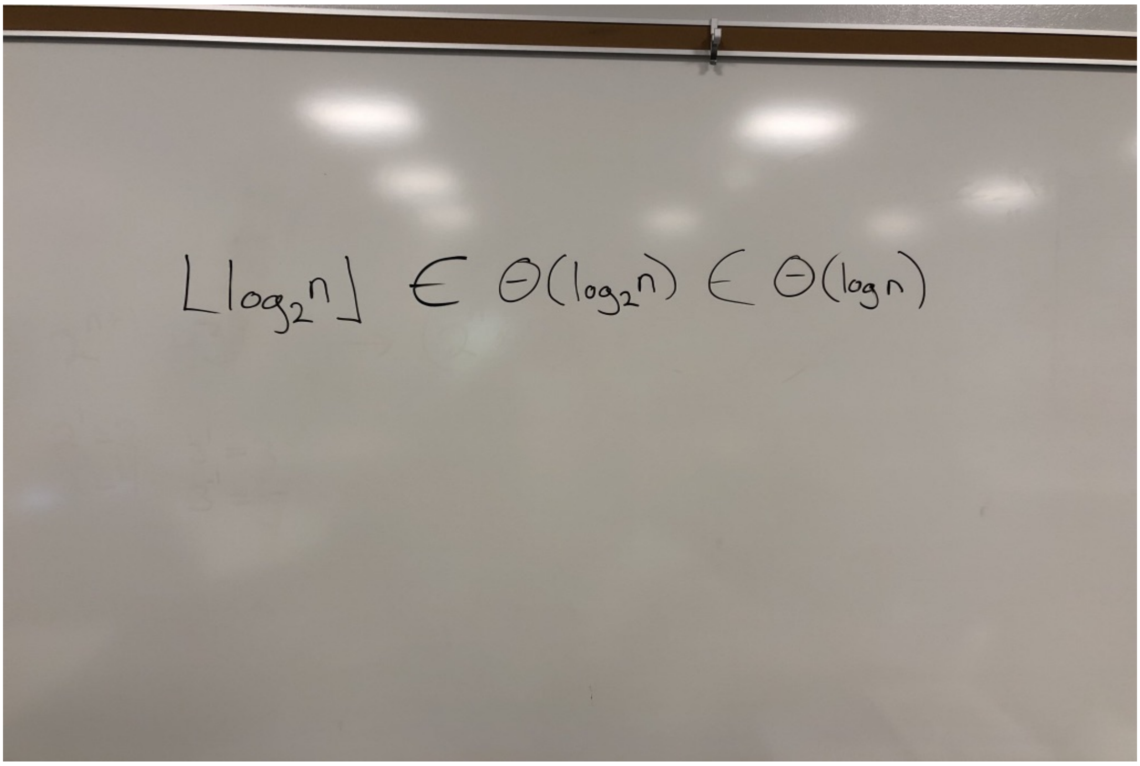

n
b) $\lim_{n\to\infty} \sqrt{10n^2+7n+3} \Rightarrow \lim_{n\to\infty} \frac{\sqrt{10n^2+7n+3}}{n} \Rightarrow \lim_{n\to\infty} \sqrt{\frac{10n^2+7n+3}{n^2}} \Rightarrow \lim_{n\to\infty} \sqrt{10+\frac{7}{n}+\frac{3}{n^2}} \Rightarrow \lim_{n\to\infty} \sqrt{10} = \boxed{\sqrt{10}}$
$\sqrt{10n^2+7n+3} \in \Theta(n)$



$\lim_{n\to\infty} 2^n \log_2(n+2)^2 + (n+2)^2 \log_2 \frac{n}{2} \Rightarrow$
$\lim_{n\to\infty} 2^n 2 \log_2(n+2) + (n+2)^2 (\log_2(n-1)) \in \Theta(n \log_2 n) + \Theta(n^2 \log_2 n) = \boxed{\Theta(n^2 \log_2 n)}$



$2^{n+1} + 3^{n-1} \rightarrow (2^n \times 2) + (3^n \times \frac{1}{3}) \rightarrow 2^n 2 + 3^n \frac{1}{3} \in \Theta(3^n)$
$2^1 = 2$ $3^1 = 3$
$2^2 = 4$ $3^{-1} = \frac{1}{3}$


$\lfloor \log_2 n \rfloor \in \Theta(\log_2 n) \in \Theta(\log n)$

4 EXERCISE 04

Problem: Compute the following sums.
(a.) 1 + 3 + 5 + 7 + ... + 999 (b.) 2 + 4 + 8 + 16 + ... + 1024 (c.) Summation (n + 1, i=3, 1) (d.) Summation (n + 1, i=3, i) (e.) Summation (n - 1, i=0, i + 1) (f.) Summation (n, j=1, 3^{j + 1}) (g.) Summation (n, i=1 Summation (n, j=1, ij)) (h.) Summation (n, i=1, 1/i(i + 1))

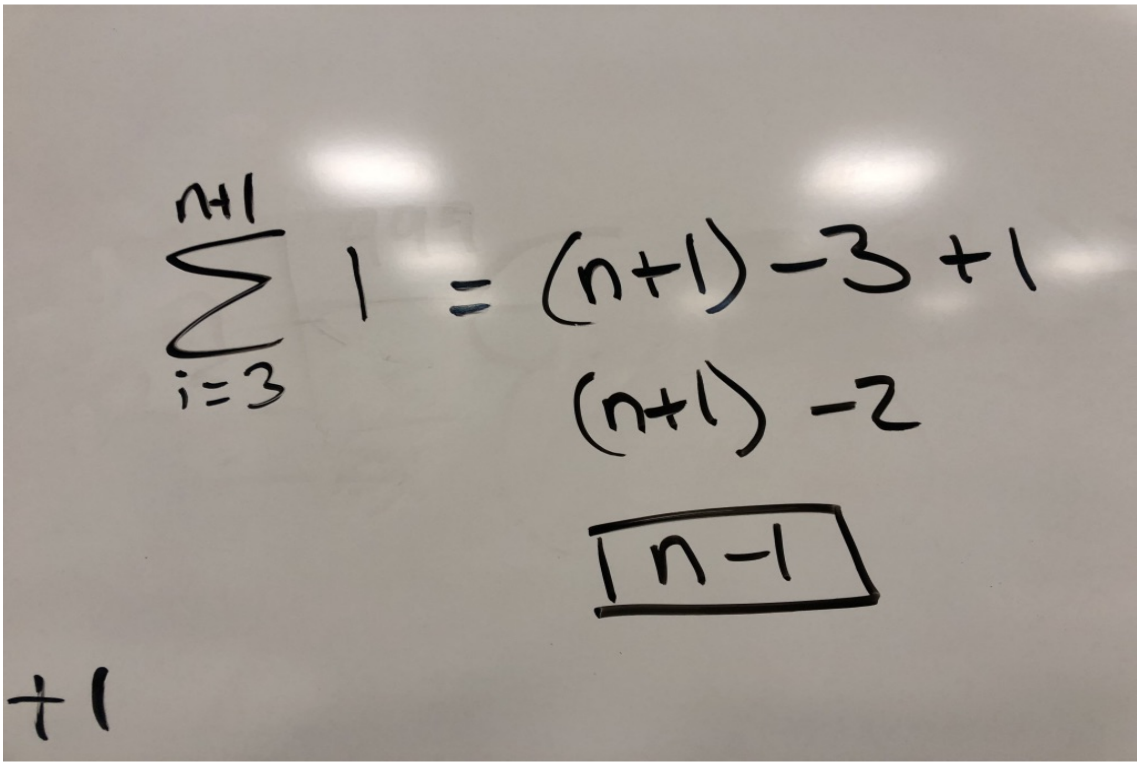
Code written in JavaScript

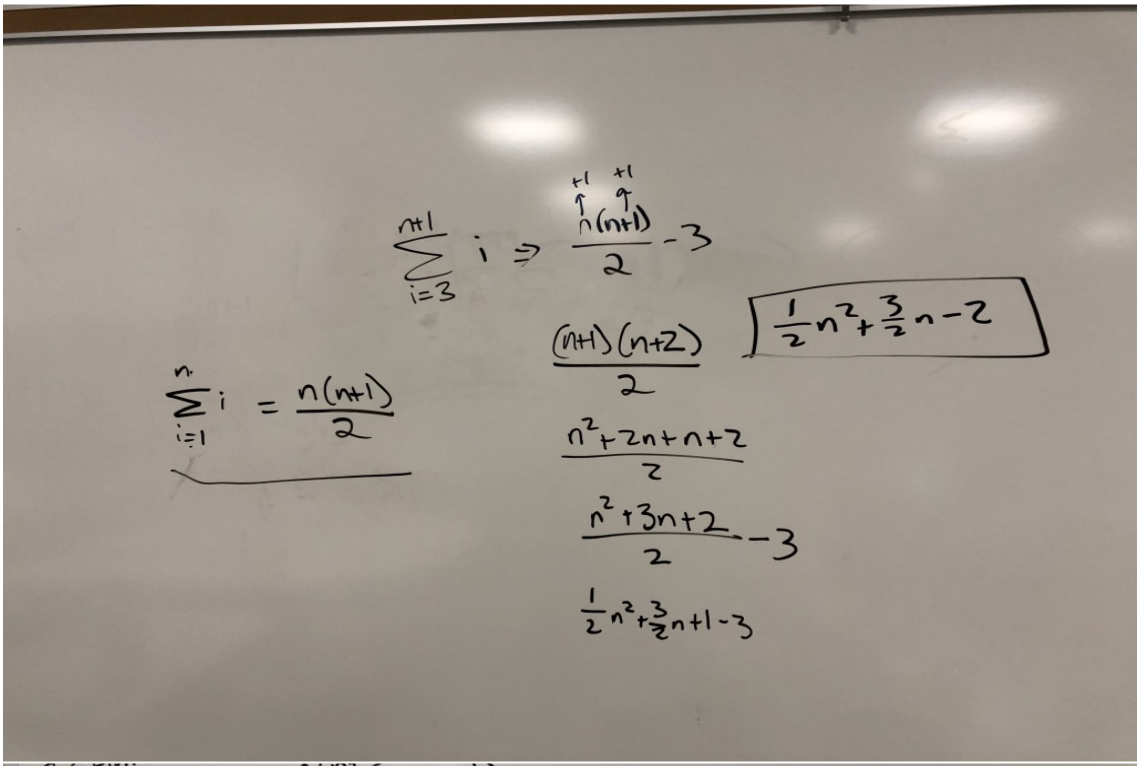
```
/**
* Exercises 2.3 - #01
* Compute the following sums.

* a. 1 + 3 + 5 + 7 + ... + 999
* b. 2 + 4 + 8 + 16 + ... + 1024
* c. Summation (n + 1, i=3, 1)
* d. Summation (n + 1, i=3, i)
* e. Summation (n - 1, i=0, i + 1)
* f. Summation (n, j=1, 3<sup>j + 1</sup>)
* g. Summation (n, i=1 Summation (n, j=1, ij))
* h. Summation (n, i=1, 1/i(i + 1))
*/
function a(currVal, max) {
    let total = 0;
    do {
        total += currVal;
        currVal += 2;
    } while (currVal <= max)
    return total;
}

function b(currVal, max) {
    let total = 0;
    do {
        total += currVal;
        currVal *= 2;
    } while (currVal <= max)
    return total;
}

function c(max) {
    let val = 0;
    for (let i=3; i <= max; i++) {
        val = i - 1;
    }
    return val;
}
```

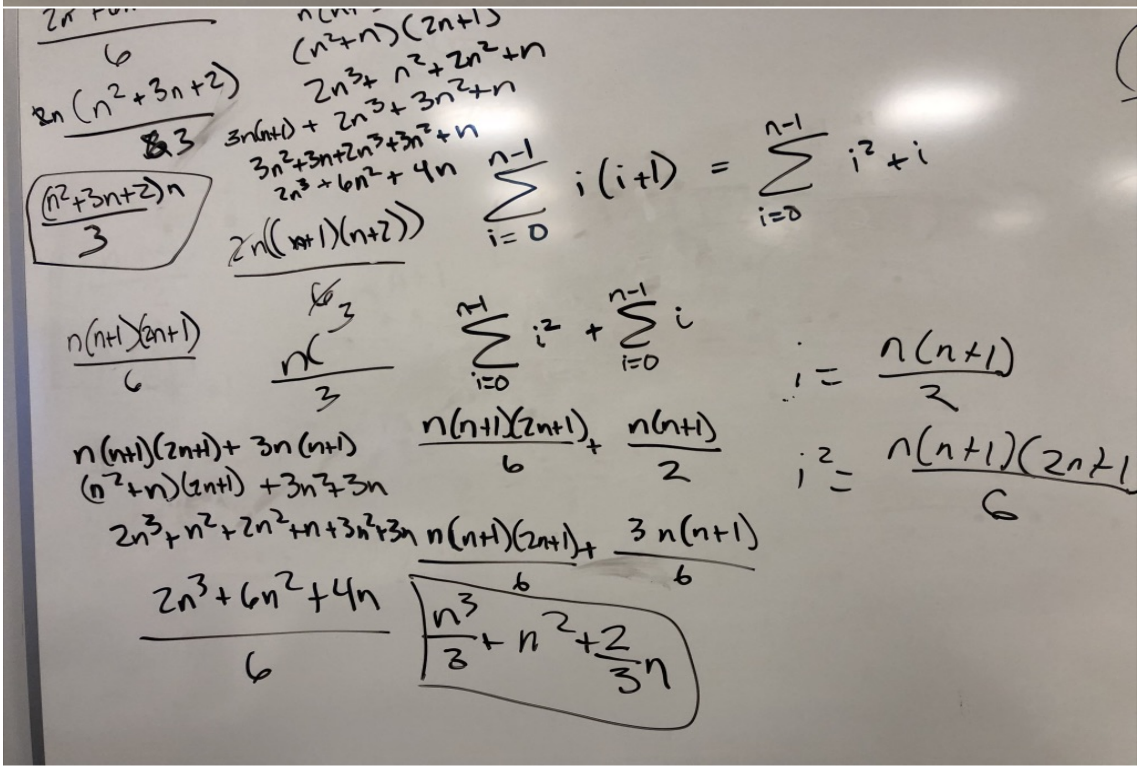
```
console.log(`a. 1 + 3 + 5 + 7 + ... + 999 = ${a(1, 999)}`);
console.log(`b. 2 + 4 + 8 + 16 + ... + 1024 = ${b(2, 1024)}`);
console.log(`c. Summation (n + 1, i=3, 1) = ${c(8)}`);
```

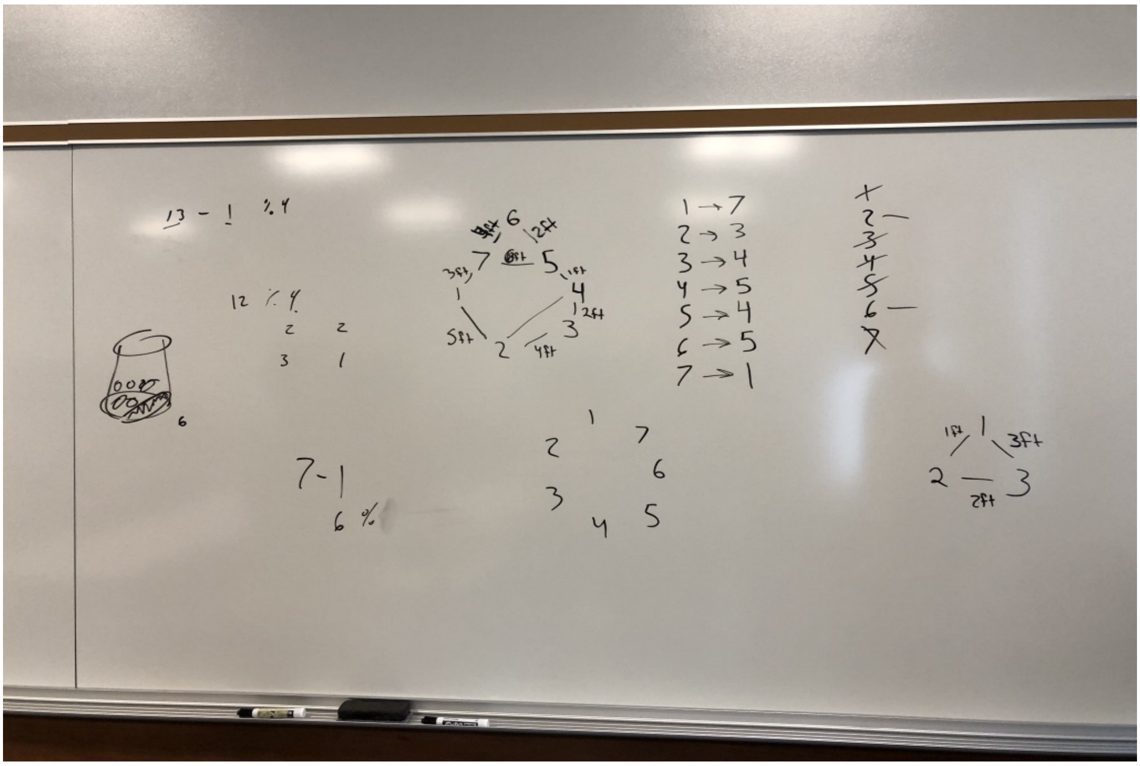
Results

```
a. 1 + 3 + 5 + 7 + ... + 999 = 250000
b. 2 + 4 + 8 + 16 + ... + 1024 = 2046
c. Summation (n + 1, i=3, 1) = 7
```




$\sum_{i=3}^{n+1} i \Rightarrow \frac{n(n+1)}{2} - 3$
$\frac{(n+1)(n+2)}{2}$
$\boxed{\frac{1}{2}n^2 + \frac{3}{2}n - 2}$
$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
$\frac{n^2+2n+n+2}{2}$
$\frac{n^2+3n+2}{2} - 3$
$\frac{1}{2}n^2 + \frac{3}{2}n + 1 - 3$


$(n^2+n)(2n+1)$
$2n^3 + n^2 + 2n^2 + n$
$2n^3 + 3n^2 + n$
$3n(n+1) + 2n^3 + 3n^2 + n$
$3n^2+3n+2n^3+3n^2+n$
$2n^3+6n^2+4n$
$\frac{n(n^2+3n+2)}{3}$
$\frac{(n^2+3n+2)n}{3}$
$\sum_{i=0}^{n-1} i(i+1) = \sum_{i=0}^{n-1} i^2 + i$
$\sum_{i=0}^{n-1} i^2 + \sum_{i=0}^{n-1} i$
$\frac{n(n+1)(2n+1)}{6}$
$i = \frac{n(n+1)}{2}$
$i^2 = \frac{n(n+1)(2n+1)}{6}$
$\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2}$
$n(n+1)(2n+1) + 3n(n+1)$
$(n^2+n)(2n+1) + 3n^2+3n$
$2n^3+n^2+2n^2+n+3n^2+3n$
$\frac{n(n+1)(2n+1)}{6} + \frac{3n(n+1)}{6}$
$\frac{2n^3+6n^2+4n}{6}$
$\frac{n^3}{3} + n^2 + \frac{2}{3}n$


1 → 7
2 → 3
3 → 4
4 → 5
5 → 4
6 → 5
7 → 1
7-1
6 %
1ft
3ft
2ft