CPSC 335-01 Project 2 Documentation

Group members: Collin Chiu (cchiu727@csu.fullerton.edu), Stephanie Becerra(stephanie_becerra_18@csu.fullerton.edu), Brian Joh (bjohdanny@csu.fullerton.edu)

Possible Improvements

While the code for all of the parts are functioning as intended, we feel that there are a few things that could be improved upon. For part 1b, we feel that the Equation 4 implementation could have been better. Our interpretation of the equation could have yielded incorrect results, but it is how we interpreted and implemented it to obtain our results. Specifically, for part 1b; rather than use given formulas to find the numbers next and behind the target number, I feel straight up just brute forcing code may take slightly longer to make but it will be easier to understand.

Pseudocode

1A) int fib(argument for number of terms in the series)

```
{
  if(number of terms < 1)
   return number of terms
  else
   return fib(previous term) + fib(two terms previous (n - 2))
}
  int userPrompt()
{
  prompt the user to input a positive non-zero number and return that number
  if(inputted number is less than 0)
    return an error and make the user re-input the number
}
```

1B)

```
// global variable
const phi = 1.61803;

// equation 3
float fSubN {
	return (pow((1 + sqrt(5)), n) - pow((1 - sqrt(5)), n)) / (pow(2, n) * sqrt(5));
}

// equation 4
void fibEqFour(int p, int n) {
	float fSubP = fSubN(p);

	for (int i = p; loop until i reaches n; increment i) {
		print fSubP;
		fSubP *= (phi)^(i - (i - 1));
	}
}
// equation 5
void fibEqFive(int p, int n) {
	float fSubP = fSubN(p);

	for (int i = p; loop until i reaches n; increment i) {
		print fSubP;
		fSubP *= phi;
	}
}
```

2) subarray(array with random values, integer containing the array length)

Integer b, e where there assigned 0, 1 respectively at the start
if (the length of the array is bigger than 2)

for (from the beginning to the length of the array)

for (one over the increment of the previous for loop to the end of the array)

if (current iterations from both for loops when summed > then the sum of)

b = first iterations
e = second for loop iteration

for (from b to e in the array)
Print the elements of the array from b to e

Results

Part 1A - fibRecursive.cpp

a) The fib function returns the nth term in the Fibonacci Sequence.

b) Output:

```
Enter the nth term of the Fibonacci series.
(Must be an integer n > 0): 15
610
```

Part 1B - fibRatio.cpp

a) userPromptP() inAsks user for non-negative integer p. If the user enters invalid input, the program asks the user for input again until the user gives a valid input.

b) Output:

```
Enter value of p.
(Must be a positive integer): 1
Enter a value of n.
(Must be a positive integer): 20
Equation 4 Fibonacci Series: 1, 1.61803, 2.61802, 4.23604, 6.85403, 11.09, 17.944, 29.0339, 46.9778, 76.0115, 122.989, 199, 321.987, 520.985, 842.969, 1363.95, 2206.91, 3
570.85, 5777.74, 9348.56,
Equation 5 Fibonacci Series: 1, 1.61803, 2.61802, 4.23604, 6.85403, 11.09, 17.944, 29.0339, 46.9778, 76.0115, 122.989, 199, 321.987, 520.985, 842.969, 1363.95, 2206.91, 3
570.85, 5777.74, 9348.56,
```

c) Based on how we implemented our fibEqFour and fibEqFive functions, our outputs are the same, but have different methods shown in the Pseudocode section above.

d) Output (Last two lines):

```
Enter value of p.
(Must be a positive integer): 3
Enter a value of n.
(Must be a positive integer): 30
Equation 4 Fibonacci Series: 2, 3.23606, 5.23604, 8.47207, 13.7081, 22.1801, 35.888, 58.0679, 93.9556, 152.023, 245.978, 397.999, 643.975, 1041.97, 1685.94, 2727.9, 4413.
82, 7141.7, 11555.5, 18697.1, 30252.5, 48949.4, 79201.7, 128151, 207352, 335501, 542851, 878349,
Equation 5 Fibonacci Series: 2, 3.23606, 5.23604, 8.47207, 13.7081, 22.1801, 35.888, 58.0679, 93.9556, 152.023, 245.978, 397.999, 643.975, 1041.97, 1685.94, 2727.9, 4413.
82, 7141.7, 11555.5, 18697.1, 30252.5, 48949.4, 79201.7, 128151, 207352, 335501, 542851, 878349,
```

Using the golden ratio to obtain Equation 5 - F_{n+1} is prone to inaccuracy due to the phi value being multiplied by each term.

Part 2 - sumArray.cpp

Sample inputs = (10, 2, -5, 1, 9, 0, -4, 2, -2);
(-7, 1, 8, 2, -3, 1);
(9, 7, 2, 16, -22, 11);
(6,1, 9, -33, 7, 2, 9, 1, -3, 8, -2, 9, 12, -4)

Sample outputs:

```
5, -1, -3, 1, 5
10, 2, -5, 1, 9
8, 2
16, -22, 11
9, -33, 7, 2, 9, 1, -3, 8, -2, 9, 12
```

Mathematical Analysis

We will be proving our codes with the use of induction. Question 1A is big O(2^n) with T(n) = 2^n + 2 because it's a recursive function that calls itself with 2 if else statements. Question 1B is big O(n) with T(n) = n + 1 because it's just one for loop that depends on n and plus one due to the if statement. Question 2 is big O(n^2) and T(n) = n^2 + 4 because there are nested for loops and 4 if and else statement

1A) $T(n) = 2^n + 2$, $O(2^n)$ 1.

2. $T(n) \leq c \cdot f(n)$

$2^n + 2 \leq c \cdot 2^n$

$c \geq 1 + \frac{2}{2^n}$

$n \geq 1$, $c = 2$ and $n_0 = 1$

3. $T(n) = 2^n + 2 = 2 + 2 = 4$

$c \cdot f(n) = 2 \cdot 2^n = 2 \cdot 2 = 4$

$4 \leq 4$, base case hold

4. $T(n) \leq c \cdot f(n)$

$2^n + 2 \leq 2(2^n)$

$n = n+1$

$2^{n+1} + 2 \leq 2(2^{n+1})$

$n = 1$

$2^2 + 2 \leq 2(2^2)$

$6 \leq 8$

so

$T(n+1) \leq c \cdot f(n+1)$

5. $2^n + 2 \in O(2^n)$

1B) $T(n) = n + 1$, $O(n)$ 1.

2. $T(n) \leq c \cdot f(n)$

$n+1 \leq c \cdot n$

$c \geq 1 + \frac{1}{n}$

$n \geq 1$, $c = 2$ and $n_0 = 1$

3. $T(n) = n + 1 = 1 + 1 = 2$

$c \cdot f(n) = 2 \cdot n = 2 \cdot 1 = 2$

$2 \leq 2$, base case holds

4. $T(n) \leq c \cdot f(n)$

$n+1 \leq 2(n)$

$n = n+1$

$(n+1)+1 \leq 2(n+1)$

$n+2 \leq 2n+2$

$n = 1$

$3 \leq 2(1)+2$

$3 \leq 4$

so

$T(n+1) \leq c \cdot f(n+1)$

5. $n+1 \in O(n)$

2) $T(n) = n^2 + 4$, $O(n^2)$ 1.

2. $T(n) \leq c \cdot f(n)$

$n^2 + 4 \leq c \cdot n^2$

$c \geq 1 + \frac{4}{n^2}$

$n \geq 1$, $c = 5$ and $n_0 = 1$

3. $T(n) = n^2 + 4 = 1 + 4 = 5$

$c \cdot f(n) = n^2 \cdot 5 = 5$

$5 \leq 5$, base case holds

4. $T(n) \leq c \cdot f(n)$

$n^2 + 4 \leq 5n^2$

$(n+1)^2 + 4 \leq 5(n+1)^2$

$n = 1$

$(2)^2 + 4 \leq 5(2)^2$

$8 \leq 20$ so $T(n+1) \leq c \cdot f(n+1)$

5. $n^2 + 4 \in O(n^2)$