Lab 2 Continued

(1)

$F(0) = 0$

$F(1) = 1$

$F(n) = F(n-1) + F(n-2)$

$F(2) = F(1) + F(0) = 1$

$F(3) = F(2) + F(1) = 2$

$F(4) = F(3) + F(2) = 3$.

Let's proof by induction

Base case: for $n=5$

$F(5) = F(4) + F(3) = 5$

$\left(\frac{4}{3}\right)^5 = 4.21$, Hence $F(5) > (4/3)^5$ ⟶ proved

Induction step:

assume for $n=k$, $F(k) > (4/3)^k$ works, we want to show that it works for $k+1$, i.e. $F(k+1) > (4/3)^{k+1}$

$F(k+1) = F(k) + F(k-1) = \left(\frac{4}{3}\right)^k + \left(\frac{4}{3}\right)^{k-1}$

$= \left(\frac{4}{3}\right)^{k-1}\left(\frac{4}{3} + 1\right)$

$= \left(\frac{4}{3}\right)^{k-1} * 7/3$

$> \left(\frac{4}{3}\right)^{k-1} * 16/9 = \left(\frac{4}{3}\right)^{k-1} * \left(\frac{4}{3}\right)^2 = \left(\frac{4}{3}\right)^{k+1}$

Hence $F(k+1) > \left(\frac{4}{3}\right)^{k+1}$ // ... proved //

Lab 2 Continued:

(2)

(a). $\lim_{n\to\infty} \frac{4^n}{2^n} = \frac{\infty}{\infty}$, we need L'Hopitals rule

Let's derivan the numerator & denominator

$\lim_{n\to\infty} \frac{\ln 4 * 4^n}{\ln 2 * 2^n} = \infty/\infty$

Let's take the k^{th} derivative

$\lim_{n\to\infty} \frac{(\ln 4)^k * 4^n}{(\ln 2)^k * 2^n} = \infty/\infty$

Hence, 4^n will never be $O(2^n)$, False//

(b) $\lim_{n\to\infty} \frac{\log n}{\log \frac{n}{3}} = \frac{\infty}{\infty}$, we need L'Hopitals rule.

let's derivate the numerator & denominator

$\lim_{n\to\infty} \frac{\frac{1}{n \ln 2}}{\frac{1}{n \ln 3}} = \frac{\ln 3}{\ln 2}$, Hence $\log n$ is $O(\log \frac{n}{3})$ //

And let's take the limit of $g(n)/f(n)$

$\lim_{n\to\infty} \frac{\log \frac{n}{3}}{\log n} = \infty/\infty$, again we need L'Hopitals rule

let's derivate the numerator & denominator

$\lim_{n\to\infty} \frac{\frac{1}{n \ln 3}}{\frac{1}{n \ln 2}} = \frac{\ln 2}{\ln 3}$, Hence $\log n$ is $\Omega(\log \frac{n}{3})$

Hence, $\log n$ is $\Theta(\log \frac{n}{3})$, True//

(c) $\lim_{n\to\infty} \frac{\frac{n}{2} \log n/2}{n \log n} = \frac{\infty}{\infty}$, we need L'Hopitals rule

derivate

$\lim_{n\to\infty} \frac{\left(\frac{1}{2} \log \frac{n}{2} + \frac{n}{2 \ln 2\, n}\right)}{\left(\log n + n/n\ln 2\right)} = \frac{\infty}{\infty}$

derivate again,

$\lim_{n\to\infty} \frac{\frac{1}{2 \ln 2 * n}}{\frac{1}{\ln 2 * n}} = \frac{1}{2}$, Hence $\frac{n}{2} \log \frac{n}{2}$ is $O(n \log n)$ //

And let's take the limit of $g(n)/f(n)$ --- Continued.

$$\lim_{n\to\infty} \frac{n\log n}{\frac{n}{2}\log\frac{n}{2}} = \frac{\infty}{\infty}$$, we need L'Hopitals rule

derivate,

$$\lim_{n\to\infty} \frac{\log n + \frac{n}{\ln 2 * n}}{\frac{1}{2}\log n/2 + \frac{n}{2\ln 2 * n}} = \infty/\infty$$

derivate again

$$\lim_{n\to\infty} \frac{1}{n\ln 2} \Big/ \frac{1}{2\ln 2 * n} = 2$$ //, Hence $\frac{n}{2}\log(\frac{n}{2})$ is $\Omega(n\log n)$

Therefore, $n/2 \log\frac{n}{2}$ is $\Theta(n\log n)$, True //

Lab 2 Continued

(3)

(a) $T(0) = 3 \quad T(1) = 3$

$T(n) = T(n-1) + 5$

For $n=2$, $T(2) = T(1) + 5 = 3+5 = 10$

$n=3$, $T(3) = T(2)+5 = 3+5+5 = 13$

$n=4$, $T(4) = T(3)+5 = 3+3+5+5 = 18$

⋮

$T(n) = 5n-2$, which is $O(n)$.

We must prove that $T(n)$ satisfies the recurrence. Proof by induction.

→ base case

$n=1$, $T(1) = 3$ ⇒ satisfied

→ assume for $n=k$, $T(k) = 5k-2$

then for $k+1$, $T(k+1) = 5(k+1) - 2 = 5k+3 = (5k-2)+5$

$= \underline{T(k)+5}$ → proved

(b) Proof of correctness

⇒ Valid recursion: Each base case is $n=0$ or $n=1$. Each self-call reduces input size by 1, so eventually base case is reached.

⇒ Base: The values 0! & 1! are correctly computed by the base case.

⇒ Recursion: Assuming recursiveFactorial $(n-1)$ correctly computes $(n-1)!$, we must show output of recursiveFactorial (n) is correct. But output of recursiveFactorial (n) is $n *$ recursiveFactorial $(n-1) = n!$, as required.

Hence, the recursion is correct

Lab 2 Continued:

(4) Psuedo Code:

Algorithm FibonacciNum(n)

Input: A non-negative Integer n.

Output: F_n which is $F_{n-1} + F_{n-2}$

if ($n=0$ || $n=1$) then return n	3
$F_1 \leftarrow 0$	1
$F_2 \leftarrow 0$	1
for $i \leftarrow 2$ to n do	$2(n-1)+n+1$
temp $\leftarrow F_2$	$n-1$
$F_2 \leftarrow F_1 + F_2$	$n-1$
$F_1 \leftarrow$ temp	$n-1$
return F_2	1

$6n+2 = T(n) \Rightarrow O(n)$

Prove the algorithm is correct.

First: we have a finite loop

The loop invariant is $I(i): F_{i-1} + F_{i-2}$

Base case: the base case for loop invariant is $i=2$

for $i=2$, $F_2 = 1$, hence base case is correct.

Induction step: assume the loop invariant works for $i=k$

$F_k = F_{k-1} + F_{k-2}$

let's proof it works for $i=k+1$

$\therefore F_{k+1} = F_k + F_{k-1}$, hence prooved.//

(5) $T(n) = T(\frac{n}{2}) + n ; T(1) = 1$

$a = 1$
$b = 2$
$c = 1$
$k = 1$

There fore, since $a < b^k$, we conclude by Master Formula that

$$T(n) = \Theta(n).$$

(6) Alporithm Zeros (A, x, lower, upper)
input: An array of length n with sorted values 0 a 1.
out put: count the number of zeros.

```
if (A[lower] = 1) then return lower.
if (A[upper] = 0) then return upper+1.

if lower > upper then return A.length

mid = (upper + lower) /2

if x = A [mid] then
    if A[mid-1] = 0 then
        return mid;
    else
    return Zeros(A, x, lower, mid-1);
else
    if(a[mid+1] = 1 ) return mid+1
 return Zeros(A, x, mid+1, upper)
```

Lab 2 Continued.

(6) - - - continued

the running time of our algorithm is

$T(n) = T(n/2) + 12$ for n a power of 2.

$T(1) = 12$

$T(2) = T(1)$

$T(4) = T(2) = 12 + 12 + 12 = 2 * 12 + 12$

$T(8) = T(4) + 12 = 3 * 12 + 12$

$\vdots$

$T(2^m) = m * 12 + 12$

assume $n = 2^m$

$T(n) = \underline{\underline{12 \log n + 12}}$

to prove $T(n)$ is $O(n)$

$\lim_{n \to \infty} \frac{12 \log n + 12}{n} = \frac{\infty}{\infty}$, we need L'Hopital's rule.

let's derivate.

$\lim_{n \to \infty} \frac{12}{n \log 2} = 0$ // $\Rightarrow$ Hence $T(n)$ is $O(n)$ //