Analysis of Algorithms

Homework 2

Thomas Schollenberger (tss2344)

September 30, 2022

Problem #1

Recall:

$$lim_{x \to \infty} \frac{f(n)}{g(n)} = c \text{ and } c \neq 0 \text{ is a constant, then } f(n) \in \theta(g(n))$$

Let $f(n)$ be T_F and $g(n)$ be φ^n

$$\lim_{x \to \infty} \frac{T_F(n)}{\varphi^n} = \lim_{x \to \infty} \frac{F_{n+1} - 1}{\varphi^n} \tag{1a}$$

$$= \lim_{x \to \infty} \frac{\frac{1}{\sqrt{5}}(\varphi^{n+1} - \hat{\varphi}^{n+1}) - 1}{\varphi^n} \tag{1b}$$

$$\tag{1c}$$

We know that φ is > 0 and $\hat{\varphi}$ is < 0. Because of that, we know that:

$$\lim_{x \to \infty} \hat{\varphi}^n = 0 \tag{1d}$$

$$= \lim_{x \to \infty} \frac{\frac{1}{\sqrt{5}}\varphi^{n+1} - 1}{\varphi^n} \tag{1e}$$

$$= \lim_{x \to \infty} \frac{\frac{1}{\sqrt{5}}\varphi^{n+1}}{\varphi^n} \tag{1f}$$

$$= \lim_{x \to \infty} \frac{1}{\sqrt{5}}\varphi \tag{1g}$$

$$= \frac{\varphi}{\sqrt{5}} \tag{1h}$$

Problem #2

The function fibItHelper implemented the recurrence $f(n; a, b)$. What is the time complexity of fibItHelper? Write down a recurrence relation $T_f(n)$ that characterizes the time complexity in terms of the number of additions (plusses) performed; then solve the recurrence exactly using iteration.

$$T_f(0) = 0 \tag{2a}$$
$$T_f(1) = 0 \tag{2b}$$
$$T_f(n+1) = 1 + T_f(n) \tag{2c}$$

Which makes $T_f(n) = n - 1$

Problem #3

Let $L : \mathbb{N}^2 \to \mathbb{N}^2$ be defined as $L(a, b) = (b, a + b)$. Then $f(n; a, b)$ can be understood as $(L^n(a, b))_1$. Prove that $n \in \mathbb{N}, L^n(a, b) = (f(n; a, b), f(n+1; a, b))$.
Base case:

$$L^0(a, b) = 1(a, b)_1 = a = f(0; a, b) \tag{3a}$$

Assume that $L^n(a, b) = (f(n; a, b), f(n + 1; a, b))$ is true, where $0 \leq k < n$, then:

$$L^{k+1}(a, b) = L_{k+1}(L_k(...(L_2(L_1(a, b))))) \tag{3b}$$
$$L^{k+1}(a, b) = L_{k+1}(f(n; a, b), f(n + 1; a, b)) \tag{3c}$$
$$L^{k+1}(a, b) = (f(n + 1; a, b), f(n + 1; a, b) + f(n; a, b)) \tag{3d}$$
$$L^{k+1}(a, b) = (f(n + 1; a, b), f(n + 2; a, b)) \tag{3e}$$

Problem #5

(a) An algorithm runs in pseudo-polynomial time if the runtime is some polynomial in the numeric value of the input, rather than in the number of bits required to represent it

(b) Yes. It is pseudo-polynomial because it scales exponentially in steps whenever you increase n by 1. This explains why it slows drastically by n = 30.

(c) No. It is a polynomial time algorithm, as it linearly scales (bits * n), which is what allows it to be extremely fast.

Problem #6

Solve the following recurrences exactly using the iteration method. In all cases, $T(0) = 0$. Answers should be expressed in terms of $T(n)$

a. $T(n+1) = T(n) + 5$

$$\begin{align} T(n+1) &= T(n) + 5 \tag{4a}\\ &= (T(n-1)+5)+5 \tag{4b}\\ &= ((T(n-2)+5)+5)+5 \tag{4c}\\ T(n) &= T(n-k)+5k \tag{4d}\\ T(n) &= 5n \tag{4e} \end{align}$$

b. $T(n+1) = n + T(n)$

$$\begin{align} T(n+1) = n + T(n) \tag{5a}\\ = n(n+T(n-1)) \tag{5b}\\ = (n+1)(n((n-1)+T(n-2))) \tag{5c}\\ T(n) = T(n-k) + \sum_{i=0}^{k}(n-i) \tag{5d}\\ T(n) = n! \tag{5e} \end{align}$$

Problem #7

Solve the following recurrences exactly using the iteration method. In all cases, $T(0) = 1$. Answers should be expressed in terms of $T(n)$

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

$$\begin{align} T(n+1) &= 2T(n) \tag{6a}\\ &= 2(2T(n-1)) \tag{6b}\\ &= 2(2(2T(n-2))) \tag{6c}\\ T(n) &= 2kT(n-k) \tag{6d}\\ T(n) &= 2n \tag{6e} \end{align}$$

b. $T(n+1) = 2^{n+1} + T(n)$

$$\begin{align} T(n+1) &= 2^{n+1} + T(n) \tag{7a}\\ &= 2^{n+1} + (2^{n} + T(n-1)) \tag{7b}\\ &= 2^{n+1} + (2^{n}(2^{n-1} + T(n-2))) \tag{7c} \end{align}$$

$$T(n) = \sum_{i=0}^{k}(2^{n-1}) + T(n-k)T(n) = 2^{n!} \tag{7d}$$

Problem #8

Solve the following recurrences exactly using the iteration method. In all cases, $T(1) = 1$. Answers should be expressed in terms of $T(n)$

a. $T(n) = n + T(n/2)$ (Assume n has the form $n = 2^m$.)

$$
\begin{aligned}
T(n) &= n + T(n/2) && \text{(8a)}\\
&= n + (n/2 + T((n/2)/2)) && \text{(8b)}\\
&= n + (n/2 + (n/4 + T((n/4)/2))) && \text{(8c)}\\
T(n) &= n + (n/2k + T(n/2k)) && \text{(8d)}\\
T(n) &= 2n - 1 && \text{(8e)}
\end{aligned}
$$

b. $T(n) = 1 + T(n/3)$ (Assume n has the form $n = 3^m$.)

$$
\begin{aligned}
T(n) &= 1 + T(n/3) && \text{(9a)}\\
&= 1 + (1 + T((n/3)/3)) && \text{(9b)}\\
&= 1 + (1 + (1 + T(((n/3)/3)/3))) && \text{(9c)}\\
m &= log_3(n) && \text{(9d)}\\
T(n) &= 1 + \log_3(n) && \text{(9e)}
\end{aligned}
$$