Assignment #1

2/11/22

1a) $2^{n+1} \in O(2^n)$ $\lim_{n\to\infty} \frac{2^{n+1}}{2^n} = \frac{2^{\infty+1}}{2^\infty} = \frac{2^\infty+2}{2^\infty} = \frac{\infty+2}{\infty} = 2$

$f(n) = 2^{n+1}$

$g(n) = 2^n$ Since the $\lim_{n\to} \frac{f(n)}{g(n)} \neq 0$, 2^{n+1} is not upper bounded by 2^n.

1b) $2^{2^n} \in O(2^n)$ $\lim_{n\to\infty} \frac{2^{2^n}}{2^n} = \ln(2)\cdot\lim_{n\to\infty}(2^{2^n}) = \ln(2)\cdot\infty = \infty$

$f(n) = 2^{2^n}$

$g(n) = 2^n$ Since the $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq 0$, 2^{2^n} is not upper bounded by 2^n.

1c) $n^{1.01} \in O(\log_2(n))$ $\lim_{n\to\infty} \frac{n^{1.01}}{\log_2(n)} = \lim_{n\to\infty} \frac{1.01n^{1.01}}{\frac{1}{n\cdot\ln(2)}} \approx 0.7\cdot n^{1.01}$

$f(n) = n^{1.01}$

$g(n) = \log_2(n)$ $= 0.7\cdot n^\infty = \infty$

Since the $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq 0$ $n^{1.01}$ is not upper bounded by $\log_2(n)$

1d) $n^{1.01} \in \Omega(\log_2 n)$ $\lim_{n\to\infty} \frac{n^{1.01}}{\log_2(n)} > 0$

$n^{1.01}$ is lower bounded by $\log_2(n)$.

1e) $\sqrt{n} \in O((\log n)^3)$ $\lim_{n\to\infty} \frac{\sqrt{n}}{(\log n)^3} = \lim_{n\to\infty} \frac{\ln(2)\sqrt{n}}{6\log_2(n)^2} = \frac{\ln^3(2)}{48} \lim_{n\to\infty} \sqrt{n} = \infty$

Since $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq 0$, $\sqrt{n}$ is not upper bounded by $(\log n)^3$

1f) $\sqrt{n} \in \Omega(\log n)^3$ Since the $\lim_{n\to\infty} \frac{\sqrt{n}}{(\log n)^3} > 0$, $\sqrt{n}$ is lower bounded by $(\log n)^3$

2b) This function is recurssive in nature. It first takes an argument x that represents the nth terms of the Fibonacci sequence to compute. Then does a conditional test where if x is less than equal to 1 it returns to x. If not, the function calls itself twice with arguments x-1 and x-2 then returns the sum of both calls.

3b) C_1: integer assignment

C_2: cost per itteration of loop

$$W(n) = 2C_1 + n \times C_2 \in O(n)$$

$$S(n) = 2C_1 + n \times C_2 \in O(n)$$

3f) C_1: Thread opperation

C_2: Conditional check

C_3: List opperation

$$W(n) = \sum_{i=0}^{\log n} \left(C_3 n + 2^i C_1\right) + C_2$$

$$W(n) \in O(n \log n)$$

$$S(n) = \sum_{i=0}^{\log n} \left(\log(n) - i\right) + C_2$$

$$S(n) \in O(\log^2(n))$$

3d)

$$W(n) = \sum_{i=0}^{n} (i \cdot C_3) + C_2$$

$$W(n) \in O(n^2)$$

$$S(n) = \sum_{i=0}^{n} (i \cdot C_3) + C_2$$

$$S(n) \in O(n^2)$$