Assignment 1

CMPS2200
Isabelle Chow

1. Asymptotic notation

$\boxed{1.a}$ Is $2^{n+1} \in O(2^n)$?

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

$2^{n+1} = c(2^n)$ set $c = 2$

$2^{n+1} = 2 \cdot 2^n$

$2 \cdot 2^n = 2 \cdot 2^n$ ✓

$\therefore 2^{n+1} \in O(2^n)$ because c can be 2 2 $\therefore$ $\boxed{\text{TRUE}}$

$\boxed{1b.}$ Is $2^{2n} \in O(2^n)$?

$2^n \cdot 2^n \leq c(2^n)$

$4^n \not\leq c(2^n)$

$\therefore$ 2^{2n} does not exist in $O(2^n)$ because no c can be found such that $4^n \leq c \cdot 2^n$ $\therefore$ $\boxed{\text{FALSE}}$

$\boxed{1c.}$ Is $n^{1.01} \in O(\log^2 n)$

$n^{1.01} = O \log_2 n$

$n^{1.01} \leq c \log_2 n$ let $c = 1$

$n^{1.01} \leq \log_2 n$

Polynomial vs grows faster than polylog

no value of n would ~~meet~~ this statement $\therefore$ $\boxed{\text{FALSE}}$

L'Hopitals rule: $\lim_{n \to \infty} \frac{n^{1.01}}{\log^2 n} = \infty$

$\log^i n \in O(n^j)$ for all i, $j > 0$

$n^j \in \Omega(\log^i n)$ for all j, $i > 0$

$\boxed{1d.}$ $n^{1.01} \in \Omega(\log^2 n)$

$n^j \in \Omega(\log^i n)$ for all j, $i > 0$

$i = 2$ $\therefore$ $i > 0$

$\therefore$ $\boxed{\text{TRUE}}$

$\boxed{1e.}$ Is $\sqrt{n} \in O(\log^3 n)$

$n^{0.5} \not\in O(\log^3 n)$

polynomial — polylog, see 1d 1c

$\therefore$ $\boxed{\text{FALSE}}$

$\boxed{1f.}$ Is $\sqrt{n} \in \Omega(\log^3 n)$

$\therefore$ $\boxed{\text{TRUE}}$ from proof of previous question because Ω is opposite to O

2a def foo in main.py - fibonacci (handwriting for submission simplicity... I have a separate code file!)

```
def foo (x)
    a = 0
    b = 0
    if x <= 1:
        return x
    else:
        a = (foo(x-1))
        b = (foo(x-2))
    return a + b
```

2b This function does the fibonacci sequence by checking if the input is 1, then adding the previous number two current number. Each number is the sum of the two preceding ones.