

**CSCI 2300 — Algo**  
**Homework 1**  
Hayden Fulle

• **Q1**

0.1. In each of the following situations, indicate whether  $f = O(g)(<=)$ , or  $f = \Omega(g)(>=)$ , or both (in which case  $f = \Theta(g)(=)$ ).

- (a)  $\Theta$
- (b)  $O$
- (c)  $\Theta$
- (d)  $\Theta$
- (e)  $\Theta$
- (f)  $\Theta$
- (g)  $\Omega$
- (h)  $\Omega$
- (i)  $\Omega$
- (j)  $\Omega$
- (k)  $\Omega$
- (l)  $O$
- (m)  $O$

• **Q2**

$$T(n) = 2^n$$

This code can be represented by  $T(0) = 1$  and  $T(n) = 1 + T(n-1) + T(n-2) + \dots + T(0)$   
 $= 1 + \sum_{i=0}^{n-1} T(i)$

Proof via induction

Base:  $T(0) = 1 = 2^0$

Induction:

$$T(n) = T(n-1) + T(n-2) + \dots + T(0) + 1$$

$$T(n-1) = T(n-2) + T(n-3) + \dots + T(0) + 1$$

$$T(n) = T(n-1) + T(n-1) = 2 * T(n-1)$$

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

therefore,  $T(n) = 2^n$  ■

• **Q3**

$$\max(f(n), g(n)) = \Theta(f(n) + g(n))$$

Growth rate is determined by the term with the greatest growth rate, and as  $n$  approaches infinity, it becomes the only term that matters. This means that as  $n$  approaches infinity, it will choose the function with the larger growth rate. When adding the  $f$  and  $g$ , the term with the largest growth rate will remain the same, so it will still have that growth rate. Since the two sides share the same term with the highest growth rate,  $\max(f(n), g(n)) = \Theta(f(n) + g(n))$

- **Q4**

a) no

b) because  $2^{2n} = 2^n * 2^n$  so  $\frac{2^{2n}}{2^n} = \frac{2^n * 2^n}{2^n} = 2^n$ , which grows towards infinity without bound as  $n$  approaches infinity.