

CS61B Extra Problems 6

I Hope You Like Math...

Using big-O notation.

- (a) For each pair of functions $f(n)$ and $g(n)$, state whether $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, or $f(n) = \Theta(g(n))$. For example, for $f(n) = n^2$ and $g(n) = 2n^2 - n + 3$, write $f(n) = \Theta(g(n))$.
- $f(n) = n$ and $g(n) = n^2 - n$
 - $f(n) = n^2$ and $g(n) = n^2 + n$
 - $f(n) = 8n$ and $g(n) = n \log n$
 - $f(n) = 2^n$ and $g(n) = n^2$
 - $f(n) = 3^n$ and $g(n) = 2^{2n}$
- (b) For each of the following functions $f(n)$, state the order of growth using Θ notation. For example, for $f(n) = n + 5$ write $f(n) = \Theta(n)$.
- $f(n) = 50$
 - $f(n) = n^2 - 2n + 3$
 - $f(n) = n + \dots + 2 + 1$
 - $f(n) = n^{100} + 1.01^n$
 - $f(n) = n^{1.1} + n \log n$
-

Code Analysis

For the pseudo-code snippet below, give the asymptotic running time in Θ notation. Assume that basic arithmetic operations (+, -, ×, and /) are constant time.

```
for  $i := 1$  to  $n$  do  
     $j := 0$ ;  
    while  $j \leq i$  do  
         $j := j + 2$ 
```

Credits:

Problems **shamelessly and completely** stolen from CS 170 Discussion 1, Fall 2015.