
Asymptotics Practice Problems

Alex Kazorian

March 9, 2017

- Order the following functions so that $f_i \in O(f_j) \iff i < j$:
 - $f_1(n) = 3^n$
 - $f_2(n) = \frac{n}{2}$
 - $f_3(n) = 2^{10000}$
 - $f_4(n) = 2^{\log_2 n}$
 - $f_5(n) = \log n$
 - $f_6(n) = n + n^2 \log n$
 - $f_7(n) = n!$
 - $f_8(n) = 1.001^n + n^3$
- Provide the tightest bound on $f(n)$ in terms of $g(n)$ by saying $f = O(g)$, $f = \Omega(g)$, or $f = \Theta(g)$.
 - $f(n) = \log_2 n$
 $g(n) = \log_3 n$
 - $f(n) = \log n^2$
 $g(n) = \log n$
 - $f(n) = n - 100$
 $g(n) = n + 10000$
 - $f(n) = 2^{1.1n}$
 $g(n) = \sum_i^n i^2$
- Find the tightest bound of the function $f(n) = x \sin x^2$ in both the worst and best case.
- Provide the worst and best runtime for the following methods in terms of N .
 - Here consider N to be the length of the array and p to be initialized to N .


```
public void int foo(int[] fighters, int p) {
    if (p != 0) {
        return fighters[p];
    }
    if (p % 2 == 0) {
        return foo(fighters, p / 2);
    }
    System.arraycopy(fighters, 0, fighters, 0, p + 1);
    fighters[p] = p;
    return foo(fighters, p + 1);
}
```