Training Problems #2 Solutions

1.

$$\text{Is } 2^{n+1} = O(2^n)? \text{ Is } 2^{2n} = O(2^n)?$$

Solution:

$2^{n+1} = O(2^n)$, but $2^{2n} \neq O(2^n)$.

To show that $2^{n+1} = O(2^n)$, we must find constants $c, n_0 > 0$ such that

$0 \leq 2^{n+1} \leq c \cdot 2^n$ for all $n \geq n_0$.

Since $2^{n+1} = 2 \cdot 2^n$ for all n, we can satisfy the definition with $c = 2$ and $n_0 = 1$.

To show that $2^{2n} \neq O(2^n)$, assume there exist constants $c, n_0 > 0$ such that

$0 \leq 2^{2n} \leq c \cdot 2^n$ for all $n \geq n_0$.

Then $2^{2n} = 2^n \cdot 2^n \leq c \cdot 2^n \Rightarrow 2^n \leq c$. But no constant is greater than all 2^n, and so the assumption leads to a contradiction.

2. Find the upper bound for $f(n) = n^2 + 1$. Provide a c and n_0.

Solution:

$n^2 + 1 \leq 2n^2, for\ all\ n \geq 1$

$Therefore\ n^2 + 1 = O(n^2)\ with\ c = 2\ and\ n_0 = 1$

3. Prove that $f(n) = 100n + 5 \neq \Omega(n^2)$

Solution:

There exists c and n_0 such that $0 \leq cn^2 \leq 100n + 5$

$100n + 5 \leq 100n + 5n = 105n\ , for\ all\ n \geq 1$

$cn^2 \leq 105n \Rightarrow n(cn - 105) \leq 0$

Since n is positive $\Rightarrow cn - 105 \leq 0 \Rightarrow n \leq 105/c$

This cannot possibly hold true for an arbitrarily large n, since c is constant.

4. Prove that $n \neq \theta(\log n)$

Solution:

$$c_1 \log n \leq n \leq c_2 \log n \Rightarrow c_2 \geq \frac{n}{\log n}, for\ all\ n\ \geq n_0$$

This cannot possibly hold true for an arbitrarily large n, since c_2 is constant.

5. What is the worst case rate of growth for the following code:

```
for (i = 1; i <= n; i=2*i)
        System.out.println(i);
```

Solution:

We start with i = 1.

At iteration 1: i = 1*2 = 2

At iteration 2: i = 2*2 = 4

At iteration 3: i = 4*2 = 8

At iteration 4: i = 8*2 = 16

At iteration 5: i = 16*2 = 32

Let j be the iteration number. We see that i is increasing at a rate of 2^j.

How many iterations does the for loop take though? We see that the for loop exits once i reaches n (because as you can see, the for loop says i <= n).

Let x be the last iteration of the for loop.

Based on the above two points, we get the following equation: $2^x = n$

Now solve for x:

$\log 2^x = \log n$

$x \log 2 = \log n$

$x = \frac{\log n}{\log 2}$

x = lg n

Thus, worst case rate of growth is O(lg n)