Stephen Chambers (smx227)
CS 858
February 3, 2016

Assignment 1

Exercise 2.1-3

for *i* = 1 **to** n
    if A[*i*] == *v*
        return i
return *NIL*

Invariant: Each element is accessed only once.
Proof: The variable *i* starts at 1. After the body of the for loop, *i* is incremented by 1. *i* is never incremented by more than one, and is never decremented.

Invariant: When *v* is found, *i* is at the correct index.
Proof: We proved that each element is accessed only once. This means that *i* must start at the beginning of the array and solely increment by 1 until it reaches the end of the array. Therefore, if A[*i*] has the value *v*, *i* is the correct index.

Invariant: If *v* is not in A, *NIL* is returned.
Proof: We proved that each element is accessed only once. We also proved that when *v* is found, *i* is at the correct index. Therefore, if *v* is not found, the for loop will terminate, and *NIL* will be returned.

Exercise 3.1-1

We are trying to prove the following:

$$c1 * (f(n) + g(n)) \leq \max(f(n), g(n)) \leq c2 * (f(n) + g(n))$$

First, prove that $\max\big(f(n), g(n)\big) = O(f(n) + g(n))$:

1. $f(n) > 0$
2. $g(n) > 0$
3. $f(n) + g(n) \geq f(n)$
4. $f(n) + g(n) \geq g(n)$
5. $f(n) + g(n) \geq max(f(n), g(n))$
6. $c2 * f(n) + g(n) \geq max\big(f(n), g(n)\big)\ with\ c2 = 1$

Next, prove that $\max\big(f(n), g(n)\big) = \Omega(f(n) + g(n))$:

1. $f(n) > 0$
2. $g(n) > 0$
3. $\max\big(f(n) + g(n)\big) \geq f(n)$
4. $\max\big(f(n) + g(n)\big) \geq g(n)$
5. $2 * \max\big(f(n) + g(n)\big) \geq f(n) + g(n)$
6. $\max\big(f(n) + g(n)\big) \geq c1 * f(n) + g(n) with\ c1 = \frac{1}{2}$

Therefore, $\max\big(f(n), g(n)\big) = \theta(f(n) + g(n))$.