

**CSC165 WINTER 2004**  
**ASSIGNMENT #4**

- (1) Prove, using cases where appropriate, that the following two Java methods compute the same function.

```
boolean f(boolean a, boolean b,
           boolean c, boolean d) {
    if (a) {
        if (b) {
            return true;
        } else if (!c) {
            return false;
        } else {
            return (c && d);
        }
    } else {
        return true;
    }
}

boolean g(boolean a, boolean b,
           boolean c, boolean d) {
    return (c && d) || b || !a;
}
```

- (2) Prove or disprove that

$$\forall i \in N, \exists j \in N, j \leq i + 1 \wedge a_j \geq i + 1$$

is true for

$$1, 0, 2, 3, 4, 5, 6, 7, \dots$$

Be sure to give a formal definition of the sequence, and use cases if appropriate.

- (3) Consider the binary search of an array with  $2^n$  elements. If the searched-for element is at index  $k$ , what does the binary representation of  $k$  tell you about how the search proceeded?
- (4) Consider running the MULT lecture code, with  $m = 25$  and  $n = 11$ . Make a table to trace the values of  $x$ ,  $y$  and  $z$ , each row containing the values at the beginning of an iteration of the loop or the end of the last iteration. Use binary representation for the values in the table.

(5) For  $a, b \in N$ , let  $m(a, b)$  be

$$\exists n \in N, a = nb.$$

For  $a, b, c \in N$ , let  $\text{gcd}(a, b, c)$  be

$$m(a, c) \wedge m(b, c) \wedge \forall n \in N, (m(a, n) \wedge m(b, n)) \rightarrow n \leq c.$$

Assume the following about mod:

$$\forall a \in N, \forall b \in N, b > 0 \rightarrow (0 \leq a \bmod b < b \wedge \exists n \in N, a = nb + a \bmod b).$$

Consider the following pseudocode for calculating the greatest common divisor:

// Precondition:  $a > b \geq 0$ .

$x = a$

$y = b$

// Loop invariant:  $x > y \geq 0 \wedge \forall n \in N, \text{gcd}(x, y, n) \rightarrow \text{gcd}(a, b, n)$ .

while  $y \neq 0$

$r = x \bmod y$

$x = y$

$y = r$

// Postcondition:  $\text{gcd}(a, b, x)$ .

- (a) Prove that the loop invariant is true at the start of the first iteration of the loop.
  - (b) Prove that if the loop invariant is true at the start of an iteration then it's true at the end of the iteration.
  - (c) Assuming that the loop terminates, and the loop invariant is true when the loop terminates, prove that the postcondition is true.
- (6) From the definitions of  $O$  and  $\Theta$ , prove or disprove each of the following:
- (a)  $(n + 3)^2 \in \Theta(n^2)$ .
  - (b)  $\log(n^2) \in O(\log n)$ .
  - (c)  $(\log n)^2 \in O(\log n)$ .
- (7) In lecture, big-O was defined by

$$O(f) = \{g : N \rightarrow R^{\geq 0} \mid \exists c \in R^+, \exists b \in N, \forall n \in N, n \geq b \rightarrow g(n) \leq cf(n)\}.$$

Sometimes the difference between  $>$  and  $\geq$  (similarly,  $<$  and  $\leq$ ) is significant. Other times it's not. Consider the following variants of big-O:

$$O_1(f) = \{g : N \rightarrow R^{\geq 0} \mid \exists c \in R^+, \exists b \in N, \forall n \in N, n > b \rightarrow g(n) \leq cf(n)\}.$$

$$O_2(f) = \{g : N \rightarrow R^{\geq 0} \mid \exists c \in R^+, \exists b \in N, \forall n \in N, n \geq b \rightarrow g(n) < cf(n)\}.$$

(a) Let  $f : N \rightarrow R^{\geq 0}$ . Does  $O_1(f) = O(f)$ ?

(b) Let  $f : N \rightarrow R^{\geq 0}$ . Does  $O_2(f) = O(f)$ ?

(c) We've been examining functions from  $N \rightarrow R^+$ . What if instead we restrict our work to functions from  $N \rightarrow R^+$ . Do part (a) again with this restriction (i.e., replacing  $R^{\geq 0}$  with  $R^+$  in the definitions of  $O$  and  $O_1$ , and in the assumption about  $f$ ).