Homework 1

CS 624, 2024 Fall

1. Consider the following algorithm for calculating a number raised to a power. The input is a real number x and a nonnegative integer n. The output is a real number r.

Algorithm 1 Power(x, n)

```
r ← 1; y ← x; p ← n
while p > 0 do
  if p is odd then
    r ← r × y
  end if
  y ← y × y
  p ← ⌊p/2⌋
end while
return r
```

The correctness property for this algorithm is the following:

$$r = x^n$$

(a) State the loop invariant for the **while** loop.

(b) Prove the correctness of the algorithm using the loop invariant.

(c) What is the running time of this algorithm? Justify your answer.

2. Consider the following algorithm for calculating the *cumulative sums* of an array. The input is an array of numbers, `A`. The output is a new array of number of the same length, `R`. (Array indexes start at 1.)

Algorithm 2 CumulativeSums(A)

```
R ← new array(length[A])
if length[A] > 0 then
  R[1] ← A[1]
end if
for j ← 2 to length[A] do
  R[j] ← R[j − 1] + A[j]
end for
return R
```

The correctness property for this algorithm is the following:

$$R[n] = \sum_{i=1}^{n} A[i] \qquad \text{for all } 1 \leq n \leq \textit{length}[A]$$

(a) State the loop invariant for the **for** loop.

(b) Prove the correctness of the algorithm using the loop invariant.

(c) What is the running time of this algorithm? Justify your answer.

3. Problem 3-4 (a, b, c, d) in the textbook (page 62).

   Let f and g be asymptotically positive functions. Prove or disprove each of the following:

   (a) $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.

   (b) $f(n) + g(n) = \Theta(\min(f(n), g(n)))$.

   (c) $f(n) = O(g(n))$ implies $\lg(f(n)) = O(\lg(g(n)))$, where $\lg(g(n)) \geq 1$ and $f(n) \geq 1$ for all sufficiently large n.

   (d) $f(n) = O(g(n))$ implies $2^{f(n)} \in O(2^{g(n)})$.

4. Problem 4-1 (a, b, f, g) in the textbook (page 107).

   Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 2$. Make your bounds as tight as possible, and justify your answers.

   (a) $T(n) = 2T(n/2) + n^4$.

   (b) $T(n) = T(7n/10) + n$.

   (f) $T(n) = 2T(n/4) + \sqrt{n}$.

   (g) $T(n) = T(n-2) + n^2$.

5. Let a binary tree be either NIL or a node with left and right attributes whose values are also binary trees. Let the mindepth function be defined as follows:

$$\text{mindepth}(t) = \begin{cases} 0 & \text{if } t = \text{NIL} \\ 1 + \min(\text{mindepth}(\text{left}(t)), \text{mindepth}(\text{right}(t))) & \text{otherwise} \end{cases}$$

   and let the countnil function be defined as follows:

$$\text{countnil}(t) = \begin{cases} 1 & \text{if } t = \text{NIL} \\ \text{countnil}(\text{left}(t)) + \text{countnil}(\text{right}(t)) & \text{otherwise} \end{cases}$$

   Prove the following: If $\text{mindepth}(t) \geq n$, then $\text{countnil}(t) \geq 2^n$.

   Hint: Use induction on n.