Homework 8 (not graded)

Due date: Mar 10, 2020, 9:30am

Objective

- Determine the time-complexity of a recursive algorithm.
- Prove the correctness of an algorithm using loop-invariants and structural induction.
- Prove the correctness of a recursive algorithm using structural induction.

Exercises

1. Recursive algorithms
   a) Write a recursive method that gets passed in a number x and a non-negative integer number n and returns x^n. The method should use divide-and-conquer by splitting the problem into the a single subproblem that requires solving $x^{\lfloor n/2 \rfloor}$ where $\lfloor z \rfloor$ is the closest integer less than or equal to z (z rounded down). How does the combine step work, i.e. haw can the solution $x^{\lfloor n/2 \rfloor}$ be used to calculate x^n? Hint: Distinguish between the case where n is even and n is odd.

   b) Specify a runtime function for your solution using a recurrence relation. For simplicity, you may assume that all powers passed in to the algorithm are powers of two, i.e. n is always of the form 2^k for some integer k.

   c) Find a closed form for your recursively defined runtime function. Again, you may assume that the function argument is of the form 2^k. If you have problems finding a closed form for your runtime function from (b), find a closed form of the following function: $t(1) = 2$ and $t(n) = t(n/2) + 2$ for $n > 1$.

   d) Prove the correctness of the closed form.

   e) Using your result from (c), determine the time complexity of the recursive algorithm.

2. (10 points)

   In the following function 2^j is the value 2^j.

   ```
   int calculate(int n)
      result = 0;
      while (j = 0; j <= n; j++)
         y = 2^j;
      return y;
   ```

   a) Trace a few iterations and determine the value of y.
   b) Determine a loop invariant.
   c) Prove that the loop invariant from (c) is correct.
   d) Given the loop invariant, determine the value that the algorithm returns.

Submission

You do not need to turn in or submit your solutions.