CS 477/677 Analysis of Algorithms

Spring 2020

Homework 2

Due date: February 11, 2020

1. (U & G-required) [30 points] Solve the following recurrences using the method of your choice.

a) $T(n) = 3\,T\left(\frac{n}{3}\right) + n^2$

b) $T(n) = 8T\left(\frac{n}{2}\right) + n^3$

c) $T(n) = 4T\left(\frac{n}{2}\right) + \sqrt{n}$

2. (U & G-required) [30 points] Consider the following recursive algorithm:

```
ALGORITHM Min1(A[0..n − 1])
//Input: An array A[0..n − 1] of integer numbers
if n = 1
    return A[0]
else temp ← Min1 (A[0..n − 2])
    if temp ≤ A[n − 1]
        return temp
    else
        return A[n − 1]
```

a) [10 points] What does this algorithm compute?

b) [20 points] Set up a recurrence relation for the algorithm’s basic operation count and solve it.

3. (U & G-required) [40 points]

(a) [15 points] Write pseudocode for a recursive algorithm for computing 2^n, where n is any nonnegative integer, based on the formula $2^n = 2^{n-1} + 2^{n-1}$.

(b) [25 points] Write a recurrence for the number of additions performed by this algorithm and **solve it using the recursion-tree method**.

4. (G-Required) [20 points] Use a loop invariant to prove that the following algorithm computes `n!`:

```
Factorial(n)
{
    i ← 1
    factorial ← 1
    while ( i ≤ n )
    {
        factorial ← factorial * i
        i ← i + 1
    }
    return factorial
}
```

Extra credit:

1. [20 points] Solve the following recurrence:

$T(n) = \sqrt{n}T(\sqrt{n}) + n$.

Hint: divide the equation by *n* throughout and then make a substitution.