King Saud University
College of Computer and Information Sciences
Department of Computer Science

Design and Analysis of Algorithms (CSC311) – Spring 2017
Instructor: Prof. Mohamed Menai

Tutorial 6 (Analysis of Recursive Algorithms)
Thu. Apr. 13th, 2017

1. Solve the following recurrence relations.
   (a) $\begin{cases} T(n) = T(n-1) + 5 & \text{for } n > 1 \\ T(1) = 0 \end{cases}$
   (b) $\begin{cases} T(n) = 3T(n-1) + 5 & \text{for } n > 1 \\ T(1) = 4 \end{cases}$
   (c) $\begin{cases} T(n) = T(n-1) + n & \text{for } n > 0 \\ T(0) = 0 \end{cases}$
   (d) $\begin{cases} T(n) = T(n/3) + 1 & \text{for } n > 1 \\ T(1) = 1 \end{cases}$
2. Consider the following recursive algorithm.

Algorithm 1 $Q(n)$ ▷ Input: A positive integer n

```
1: if n = 1 then
2:     return 1
3: else
4:     return Q(n − 1) + 2 ∗ n − 1
5: end if
```

   (a) Set up a recurrence relation for this functions values and solve it to determine what this algorithm computes.
   (b) Set up a recurrence relation for the number of multiplications made by this algorithm and solve it.
   (c) Set up a recurrence relation for the number of additions/subtractions made by this algorithm and solve it.
3. (a) Design a recursive algorithm for computing 2^n for any nonnegative integer n that is based on the formula: $2^n = 2^{n-1} + 2^{n-1}$.

(b) Set up a recurrence relation for the number of additions made by the algorithm and solve it.

(c) Draw a tree of recursive calls for this algorithm and count the number of calls made by the algorithm.

(d) Is it a good algorithm for solving this problem?