Alternative HW1, CS430, Spring 2016

Instructor: Professor Edward M. Reingold
TA: Jing Zhao, Junze Han & Taeho Jung

January 22, 2016

#1 Use mathematical induction to show that when n is an exact power of 3, the solution of the recurrence

$$T(n) = \begin{cases} 9 & \text{if } n = 3 \\ 6T(n/3) + \frac{1}{3}n^2 & \text{if } n = 3^k, \text{ for } k > 1 \end{cases}$$

is $T(n) = n^2$.

#2 Write a recurrence for the running time of binary search in a sorted array $A[1 \cdots n]$.

#3 Design and describe an algorithm to calculate $A_{m \times n} \times A_{n \times o}$ where $A_{m \times n}$ has m rows and n columns and $A_{n \times o}$ has n rows and o columns, and analyze the time complexity in terms of Θ (Big-Theta) notation.

#4 Rank the following functions by the order of growth: that is, find an arrangement $g_1, g_2, \cdots, g_6$ of the functions satisfying $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \cdots, g_5 = \Omega(g_6)$. Mark if you find two functions have the same growth rate. **You must justify your answers to get credits.**

** e is the base of natural logarithm.

$$2^{\lg n} \qquad \sqrt{2}^{\lg n} \qquad \lg n^2 \qquad \sqrt{n}(n/e)^n \qquad (n+1)! \qquad n^n$$

#5 Solve the following recurrence using two ways: first with the master theorem on page 9 in the January 13 notes, and then using secondary recurrences on pages 15-16 in the January 13 notes.

$$T(n) = 2T(n/2) + n$$