Emily Campbell

CS142 – Intro to Comp Sci II

December 2, 2015

Assignment 10

Question 1: Below are two statements of the form f(n) = Θ(g(n)). Prove that each of these statements is true by finding, in each case, positive constants a, b, and n0 such that a\*g(n) < f(n) < b\*g(n) for every

n > n0. Justify your answers.

1. n2 + 5n – 3 = Θ(n2)

In this case, g(n) = n2. If we set n0 equal to 0, f(n) = -3, and there is no positive coefficient that can make g(n) equal to f(n). Therefore n0 must be greater than 0, so we will make n0 equal to 1. For every n > 1, 5n – 3 will be a positive constant C, so f(n) will be equal to n2 + C, which is greater than n2. So for every n > 1, 1\*g(n) < f(n).

With n0 = 1, we see that f(n) = 3 = 3n2. As the values of n increase, f(n) stays below 3n2 (example: f(2) = 11 < 3(2)2, f(3) = 21 < 3(3)2, etc). Therefore we can set b equal to 3. So to summarize, our values for a, b, and n0 are **1**, **3**, and **1** respectively.

1. n2 – n = Θ(n2)

In this case, g(n) = n2. If we set n0 equal to 1, f(n) = 0, and there is no positive coefficient of g(n) that can make it equal to f(n). Therefore we will say n0 = 2. Because n will always be a positive value, f(n) will always be less than n2, so we can set b equal to 1.

We can say that n = (1/n)\*n2, and thus rewrite f(n) as n2 – (1/n)\*n2. As n increases, (1/n) decreases, so (1/n) will be largest when n is at its lowest value, n0 = 2. At that value,

f(n) = (1/2)\*n2; this is the lowest value of f(n), so we can say that (1/2)\*n2 < n2 for every n > n0. Thus, a = 1/2. To summarize, our values for a, b, and n0 are **1/2**, **1**, and **2** respectively.

Question 2: What is the asymptotic running time of each of the following algorithms, as a function of *n*? Don’t forgot to simplify and use the Θ notation. Justify your answers.

(a) for i = 1 to n

for j = 1 to n\*n

print ‘\*’

Starting with the inner for loop, we see that the body of the loop will have a linear run time; that is, it will take the same amount of time to execute the body of the for loop for each element. Because the inner for loop indexes from 1 to n\*n (a.k.a. 1 to n2), the run time of the inner loop will be proportional to n2. The outer loop indexes from 1 to n, so its run time will be equal to n\*(body run time), a.k.a. n\*n2 or n3. Therefore, we can say the overall run time is **Θ(n3)**.

(b) for i = 1 to 2\*n

for j = n to n+5

print ‘\*’

The body of the inner loop will take the same amount of time to run for each element; we will say that this constant run time is equal to C. The inner loop executes 6 times for every value of n, so the inner loop will have a run time of 6\*(body run time) or 6C. The outer for loop indexes from 1 to 2n, so its run time is equal to 2n\*(body run time) a.k.a. 2n\*(6C) or n(12C). After removing the coefficient and finding the dominant term, we get a run time of **Θ(n)**.