# COMP 363 - Algorithms: Homework #NUM

Due on DUE DATE

*Leo Irakliotis*

**Nathan Hogg**

# Problem 1

The goal of this assignment is to prove the time complexity of the mergesort algorithm. We are given the base complexity formula as:

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

In this case, $T(n)$ is the time it takes to mergesort an array of size $n$, and $f(n)$ is the time it takes to assemble two partial solutions of size $n/2$ to a sorted array with $n$ elements. Generally, $f(n) \approx n$.
For an array with $n = 8$, we can write:

$$T(8) = 2T(4) + f(8)$$

Then, $T(4) = 2T(2) + f(4)$; and $T(2) = 2T(1) + f(2)$; and finally $T(1) = f(1)$. Using the last finding, we can solve backwards:

T(8) = 2T(4) + f(8)
= 2(4f(1) + 2f(2) + f(4)) + f(8)
= 8f(1) + 4f(2) + 2f(4) + f(8)