University of Waterloo
CS 466 — Advanced Algorithm
Spring 2013
Problem Set 9
Siwei Yang - 20258568

1. [14 marks: Mergesort] Consider Mergesort when n is not (necessarily) a power of 2. The method works by (recursively) sorting a subarray of size $\frac{n}{2}$ and one of size $\frac{n}{2}$ and then merging them in n-1 comparisons. A segment of length 1 requires 0 comparisons.

   - (a) [2 marks] Give a recurrence relation that describes the number of comparisons used, in the worst case, by this method.

$$C(n) = \begin{cases} C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + n - 1, & \text{if } n > 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

   - (b) [4 marks] Prove that n-1 comparisons are necessary (i.e. you cannot do it in fewer), in the *worst case* for this merge step
   - (c) [4 marks] Prove that Mergesort, as described above, takes $n * \lceil \lg n \rceil - 2^{\lceil \lg n \rceil} + 1$ comparisons in the worst case.
   - (d) [4 marks] The *expected* number of comparisons for this method (over all possible permutations of the input) is a little $\theta(n)$ better. Prove it. (You do not have to deal with the exact constant in this $\theta(n)$ term.)