Istanbul Technical University
Faculty of Computer and Informatics

BLG335E Analysis of Algorithms I
Project 1

Cem Yusuf Aydoğdu
150120251

14.09.2015

a. Bubble sort is consist of two loops. Assuming length of array is n, outer loop runs n times, and inner loop runs $n - i$ times, where i is equal to iterator of outer loop.

$$f(n) = \sum_{i}^{n} (n - i) = \frac{n^2}{2} - \frac{n}{2}$$

An asymptotic upper bound can be selected as $g(n) = n^2$, which satisfies $f(n) = \frac{n^2}{2} - \frac{n}{2} \leq c\, g(n)$ for all $n \geq n_0$, where c and n_0 are constants and $c > 0$ and $n_0 > 0$.
Selecting $n_0 = 1000$ and $c = 1$, $f(1000) = 0.013769 < 1000^2$

Merge sort is consist of recursive calls of itself to divide the problem into two parts, and some operations to sorting and collecting divided parts.

$$T(n) = \begin{cases} 1, & n < 1 \\ 2T\left(\frac{n}{2}\right) + n, & n \geq 1 \end{cases}$$

An asymptotic upper bound can be selected as $n \log_2 n$. Using Master Theorem, $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ where $a = 2, b = 2, f(n) = n$.

$f(n) = \theta\left(n^{\log_2 2}\right) = \theta\,(n)$ is true. Then $T(n) = \theta(n^{\log_2 2} \log_2 n)$

b. Completion times of algorithms for different array sizes are shown below.

N / Algorithm	1000	10000	1000000	1000000
Bubble Sort	0.013769	0.691353	72.5555	6905.35
Merge Sort	0.000893	0.006381	0.034509	0.393132

c. It is clear that merge sort is always better than bubble sort for given array size conditions. This plot below also proves that n^2 increases faster than $n \log_2 n$, especially after $N = 100000$.