Analysis of Algorithms: Assignment 5

Due date: February 8 (Thursday)

Problem 1

Determine asymptotic upper and lower bounds for each of the following recurrences. Make your bounds as tight as possible.

- **(a)** $T(n) = 3 \cdot T(n/27) + n$
- **(b)** $T(n) = 3 \cdot T(n/27) + \sqrt[3]{n}$
- **(c)** $T(n) = T(n/6) + T(n/3) + T(n/2) + n$
- **(d)** $T(n) = T(n-1) + 1/2^n$
- **(e)** $T(n) = 2 \cdot T(\sqrt{n}) + 1$

Problem 2

The standard complexity analysis of $\textsc{Merge-Sort}(A, p, r)$ is based on the assumption that, when making a recursive call, we pass the array $A[1..n]$ by a pointer, which means that parameter passing takes constant time. If a programming language does not allow passing an array by a pointer, we may have two other options:

(a) Pass the array $A[1..n]$ by copying all its elements, which takes $\Theta(n)$ time.

(b) Pass the array by copying only the elements used in the called function; then, the procedure has to copy the $A[p..r]$ segment of the array, which takes $\Theta(r - p + 1)$ time.

For each of these two options, write a recurrence for the running time of Merge-Sort and give an asymptotic upper and lower bound for the recurrence.