MAT-72006 Advanced Algorithms and Data Structures
September 15, 2016
HW 2: 3 Growth of Functions, 4 Divide-and-Conquer

1.

Prove Theorem 3.1:

For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

2.

Prove the following statements.

(a) $n^2 + 7 = O(n^2)$
(b) $4n^3 + n^2 = \Omega(n^3)$

(Hint: you might find it useful that, for positive functions f and g, it holds that $f(n) = \Omega(g(n))$ if and only if $\lim_{n\to\infty} \dfrac{f(n)}{g(n)} > 0$, given the limit exists).

3.

Explain why the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

4.

Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.

5.

Suppose we have two algorithms, $\mathcal{A}$ and $\mathcal{B}$, for solving some problem (say multiplying two $n \times n$ matrices, or sorting an array n of integers). Let n denote the input size of the problem, and let $T_{\mathcal{A}}(n)$ and $T_{\mathcal{B}}(n)$ denote the number of steps taken by algorithms $\mathcal{A}$ and $\mathcal{B}$, respectively, on inputs of size n.

It is known that $T_{\mathcal{A}}(n) = O(n^3)$ and $T_{\mathcal{B}}(n) = O(n^5)$. Is it true that for sufficiently large n, algorithm $\mathcal{A}$ performs less steps than algorithm $\mathcal{B}$? Why or why not?