Homework 1[*]

Data Structures
Fall 2019 CS203@IITG

---

(1) Prove the transitivity, reflexivity, symmetry, and transpose symmetry relations of the asymptotic notation. (Refer to pages 51-52 of [CLRS].)

(2) Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$, where $a_0, a_1, \ldots, a_n$ are real numbers with $a_n \neq 0$. Then show that $f(x)$ is of order x^n when $a_n \neq 0$.

   Determine whether $f(x)$ is of order x^n when $a_n = 0$.

(3) Write the pseudocode for summing the integers stored in a two-dimensional C array. The pseudocode is permitted to use the primitives supported by word-RAM model of computation only.

   Argue that the pseudocode is indeed an algorithm.

   Further, give the asymptotic worst-case tight bound of the time taken and the asymptotic worst-case tight bound of the space used by the algorithm.

(4) Provide the case analysis to show the code in page 240 of [CLRS] for inserting a node into a doubly linked list that uses sentinel is correct.

   When each node i of the doubly linked list L (without sentinel) is augmented with the number of nodes that are after i in L, for inserting a node into L, determine whether a sentinel helps to improve the efficiency in practice.

(5) Give the pseudocode where the stack data structure is implemented with a queue data structure and analyze the worst-case asymptotic time and work-space complexities of the resultant push and pop operations of the resultant stack. [Assume queue is implemented with a static array.]

(6) Give the pseudocode where the queue data structure is implemented with a stack data structure and analyze the worst-case asymptotic time and work-space complexities of the resultant enqueue and dequeue operations of the resultant queue. [Assume stack is implemented with a static array.]

(7) While considering the precedence and associativity rules for $(,), +, -, *, /, \hat{}$ operators in C, argue that the algorithm given to convert the infix expression to postfix form is correct.

(8) Give all the details of the deque (doubly-ended queue) data structure (possibly with simple enough pseudocode) that gets implemented using dynamic arrays.