Data Structure Homework #1

Due Oct 21, 2021 before class.

Collaboration policy: You can discuss the problem with other students, but you must obtain and write the final solution by yourself. Please specify all of your collaborators (name and student id) for each question. If you solve some problems by yourself, please also specify "no collaborators". Homeworks without collaborator specification will not be graded.

Problem 1 (10%)

Prove or disprove the following statements. You may only use the definition of asymptotic notations.

1. $3n^2 + n \in O(n^2)$.
2. If $f(n) \in O(n^2)$, then $f(n) \in o(n^3)$.

Problem 2 (10%)

Sort the following functions into a list $f_1, f_2, \ldots$ such that $f_i = \Omega(f_{i+1})$ for all i. Identify all pairs of functions $f(n)$, $g(n)$ such that $f(n) = \Theta(g(n))$. No explanation is needed.

$8^{\log_2 n}$	$n!$	$n^{0.01}$	2^{2n}	n^3+5n^2	$\log_2 n$
$\sqrt{n}+3$	$\ln n$	$(\log_2 n)!$	n^n	$(0.5)^n$	3

Problem 3 (10%) Given n elements stored in a singly linked list. In addition, the address of the i-th element is provided.

1. How much time does it take to add a new element between the i-th and the $(i+1)$-th element? (in Θ notation)
2. How much time does it take to add a new element between the $(i-1)$-th and the i-th element? (in Θ notation)

(Explain your answer briefly)

Problem 4 (15%) A d-ary max heap is the same as a binary max heap except that each node has d children instead of two.

1. Design a method to store a d-ary max heap in an array. Similar to binary heaps, you may not use any extra pointers. Please explain the parent-child relationship clearly.
2. Design an efficient method to implement the Insert and Extract-MAX operations. Analyze the running time (in terms of d and the number of elements n) of these two operations.

Problem 5 (20%)

Prove or disprove the following statements. You may only use the definition of asymptotic notations.

1. Let $f_1(n), f_2(n), \ldots, f_i(n), \ldots$ be an infinite series of functions. $f_i(n) \in O(n)$ for all i. Let $g(k) = \sum_{j=1}^{k} f_j(j)$ (i.e. $g(1) = f_1(1), g(2) = f_1(1) + f_2(2)$, and so on), then $g(n) \in O(n^2)$.
2. Let $f(n) \in O(n)$. Let $g(k) = \sum_{j=1}^{k} f(j)$ (i.e. $g(1) = f(1), g(2) = f(1) + f(2)$, and so on), then $g(n) \in O(n^2)$.

Problem 6 (20%)

1. Show how to implement a queue using two stacks. Analyze the (worst-case) running time of the queue operations.
2. Show how to implement a stack using two queues. Analyze the running time of the stack operations.

Problem 7 (15%) An array of n elements is almost sorted if and only if every element is at most k spots away from its location in a sorted array. Design an algorithm which sorts an almost sorted array in $O(n \log k)$ time. Briefly justify the correctness and analyze the running time.