Instructor Name: Siddhartha Nath
Email: s.nath@northeastern.edu

Northeastern University

Program Structure and Algorithms (INFO 6205)
Midterm Exam – 100 points

Student NAME:

Student ID:

Question 1 (30 points)**.** *Please provide short and concise answers and reasons for the following.*

(a) *[4 points] Express the function* $n^3/1000 + 100n^2 - 100n + 3$ *in* Θ*-notation.*

(b) *[5 points] Suppose you have two graph algorithms for the same problem. Alg1 has runtime* $O(|V|^{3/2})$ *and Alg2 has runtime* $O(|E|)$*. Which of the following is likely true? Choose the* **best** *option.*

(i) Alg1 is faster on most graphs.

(ii) Alg2 is faster on most graphs.

(iii) Alg1 is faster on sparse graphs, slower on dense graphs.

(iv) Alg2 is faster on sparse graphs, slower on dense graphs.

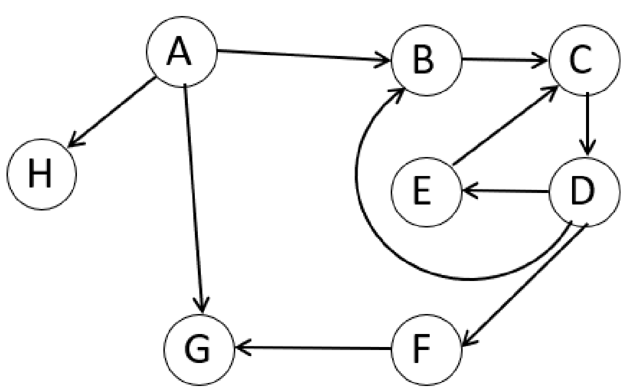
(c) *[8 points] What is the solution for* $T(n)$ *that is given by the recurrence* $T(n) = 3T(n-1) + 1$ *with* $T(0) = 1$*? Please state the solution in big-O notation and explain your solution clearly to receive full credit.*

(d) *[6 points] There is a tree where the left subtree contains 1000 nodes, and the right subtree contains 100 nodes. For each of preorder, inorder, and postorder traversals, how many nodes are processed before the root?*

Preorder: ________ *; Inorder:* ________ *; Postorder:* ________

(e) *[7 points] Given a sequence of numbers:* $19, 6, 8, 11, 4, 13, 5, 27, 43, 49, 31, 25$*, draw a* **binary search tree** *by inserting the above numbers from* **left to right**.

Question 2 (15 points)**.** *Consider the directed graph* G *below.*



(a) *[5 points] Please draw all edges of the reverse graph G^R, then execute* **depth-first search on** G^R*, starting at vertex A, and clearly write the pre and post labels next to each vertex. Break all ties lexicographically (i.e, according to alphabetical order).*

(b) *[5 points] List the strongly connected components (SCCs) of G, in the order that they are found using the algorithm given in class.*

(c) *[5 points] Draw the metagraph of strongly connected components (SCCs) for the above graph G.*

Question 3 (15 points)**.** *Consider the following three divide-and-conquer algorithms.*

- ***Algorithm A*** *when run on an input of size n solves three recursive subproblems of size $n/3$ in addition to $\Theta(n)$ overhead work.*
- ***Algorithm B*** *when run on an input of size n solves three recursive subproblems of size $n/2$ in addition to $\Theta(n^2)$ overhead work.*
- ***Algorithm C*** *when run on an input of size n solves three recursive subproblems of size $n/9$ in addition to $\Theta(1)$ overhead work.*

(a) *[6 points] Please express the recursion function of each algorithm as $T(n) = aT(n/b) + f(n)$. Please clearly indicate what are a, b and $f(n)$ to receive full credit.*

(b) *[6 points] What are the asymptotic runtimes of these three algorithms?*

(c) *[3 points] Please rank the algorithms from the fastest (1) to the slowest (3).*

Question 4 (20 points)**.** *Let $G = (V, E)$ be a directed graph where every vertex is assigned a real number. We wish to compute for each vertex $v \in G$ the largest number assigned to any vertex reachable from v.*

(a) *[8 points] Describe an algorithm in English that runs in $O(|V|+|E|)$ for this problem if G* **is strongly connected.**

(b) *[12 points] Describe an algorithm in English that runs in $O(|V| + |E|)$ for this problem if G* **is a DAG**.

Question 5 (20 points)**.** *Suppose you are given an array $A[1:n]$ of integers. We say that an element $A[i]$ is the* **peak** *if it is not smaller than its neighbor(s). That is,*

- *if $i \neq 1, n$, $A[i] \geq A[i-1]$ and $A[i] \geq A[i+1]$,*
- *if $i = 1$, $A[1] \geq A[2]$, and if $i = n$, $A[n] \geq A[n-1]$.*

For example, if $A = [4, 3, 9, 10, 14, 8, 7, 2, 2]$, the peaks are at $A[1] = 4$, $A[5] = 14$, and $A[9] = 2$. That is, the peaks appear at indices $i = 1, 5, 9$.

The task is to design an efficient algorithm to find **any** *peak in* A*. In the example, your algorithm can return any* $A[i]$ *from* $i = 1, 5, 9$*.*

(a) *[5 points] Can you describe a brute-force algorithm in English to find any peak element? What will be the running time of such an algorithm?*

(b) *[6 points] Can you describe a divide-and-conquer algorithm in English to find any peak?*

(c) *[5 points] Please write pseudocode of your D/Q algorithm in (b).*

(d) *[4 points] Please write the running time of your pseudocide in (c) as recursion function* $T(n)$*. What is the running time in* $O(\cdot)$ *or* $\Theta(\cdot)$*?*