Am I ready for this?

ONID number ONLY (no name): ______________________________

1. For each of the following, indicate whether $f = O(g)$, $f = \Omega(g)$ or $f = \Theta(g)$. If $f = \Theta(g)$, then only indicating this will give you full marks. **(7 points)**

	$f(n)$	$g(n)$	answer
(a)	$3n + 6$	$10000n - 500$	
(b)	$n^{1/2}$	$n^{2/3}$	
(c)	$\log(7n)$	$\log(n)$	
(d)	$n^{1.5}$	$n \log n$	
(e)	$\sqrt{n}$	$(\log n)^3$	
(f)	$n2^n$	3^n	
(g)	$7^{\log_4 n}$	n^2	

answer

(a) $3n + 6 = \Theta(10000n - 500)$

(b) $n^{1/2} = O(n^{2/3})$

(c) $\log(7n) = \Theta(\log(n))$

(d) $n^{1.5} = \Omega(n \log n)$

(e) $\sqrt{n} = \Omega((\log n)^3)$

(f) $n2^n = O(3^n)$

2. Solve the following recurrence relations (assume reasonable base cases). **(3 points)**

   (a) $T_A(n) = 7\,T_A(n/7) + O(n)$

   (b) $T_B(n) = 8\,T_B(n/2) + O(n^4)$

   (c) $T_C(n) = T(n-1) + O(1)$

(a) $T_A(n) = O(nlogn)$
(b) $T_B(n) = O(n^4)$
(c) $T_C(n) = O(n)$

3. For each of the following, choose among the following answers. **(10 points)**

$$O(\log n),\ O(n),\ O(n \log n),\ O(n^2),\ O(2^n)$$

(a) number of leaves in a depth-n balanced binary tree

(b) depth of an n-node balanced binary tree

(c) number of edges in an n-node tree

(d) worst-case run time to sort n items using merge sort

(e) maximum number of matched pairs in a matching between two sets of n items

(f) number of distinct subsets of a set of n items

(g) number of bits needed to represent the number n

(h) time to find the closest pair of points among n points in Euclidean space by enumeration

(i) time to insert n items into a binary heap data structure

(j) time to find the third biggest number in a set of n numbers

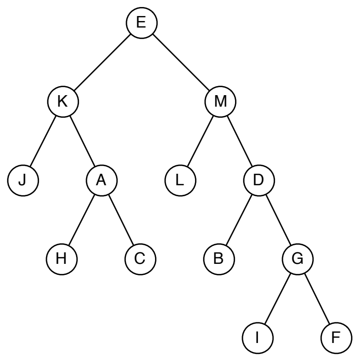
number of leaves in a depth-n balanced binary tree	$O(2^n)$
depth of an n-node balanced binary tree	$O(\log n)$
number of edges in an n-node tree	$O(n)$
merge-sort worst-case running time	$O(n \log n)$
maximum number of matched pairs in a matching between two sets of n items	$O(n)$
number of distinct subsets of a set of n items	$O(2^n)$
number of bits needed to represent the number n	$O(\log n)$
time to find the closest pair of points among n points in Euclidean space by enumeration	$O(n^2)$
time to insert n items into a binary heap data structure	$O(n \log n)$
time to find the third biggest number in a set of n numbers	$O(n)$

4. I traversed a complete (not necessarily balanced), undirected binary tree with 13 nodes using depth first search and found an ordering of the nodes by their pre-order (the order in which nodes are first visited) and post-order (the order in which nodes are last visited):

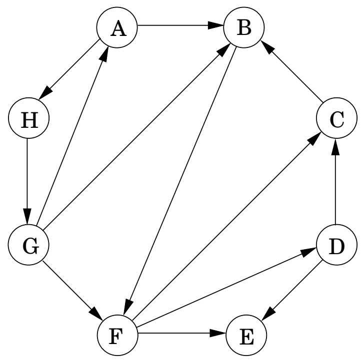
   pre-order E K J A H C M L D B G I F

   post-order J H C A K L B I F G D M E

   But then I lost the tree. Can you help me reconstruct the tree? Draw the tree that results in these orderings. Recall that a complete binary tree is one in which every node has either 2 children or no children. That is, no node has only one child. **(1 point)**



5. Consider the following graph G.



(a) List the strongly connected components of G.

(b) What is the minimum number of edges that must be added to G to make G strongly connected.

(c) Draw a graph H representing the connectivity between the strongly connected components.

(d) Give one topological ordering of the strongly connected components of G (the vertices of H).

(e) How many different topological orderings of the vertices of H (the strongly connected components of G) are there?

(5 points)

(a) $\{A, G, H\}, \{B, C, D, F\}, \{E\}$

(b) 1

(c) $\{A, G, H\} \rightarrow \{B, C, D, F\} \rightarrow \{E\}$

(d) 1

6. Let G be a connected undirected graph on n vertices. Suppose you find a breadth-first search tree T_B starting from node s and a depth-first search tree T_D also starting from s. Surprise! We happen to find out that $T_B = T_D$!

   True or false: G has $n - 1$ edges. **(1 point)**

   True.