Name: _____

CS343: Analysis of Algorithms Midterm, Sp 08

Score:	1.	(max 20)	5.	(max 15)
	2.	(max 20)	6.	(max 10)
	3.	(max 15)	7.	(max 10 Xtra)
	4.	(max 15)		
	Total:		(max 105)	percent:

This exam is closed book. Calculators are not allowed.

1. (20 pts total) Complexity

   (a) (.5 pts each, 15 pts) Fill in the following table. That is, state (yes or no) whether A is $O(B)$, $o(B)$, $\Omega(B)$, $\omega(B)$, and/or $\Theta(B)$. Assume c and k are positive constants.

A	B	O	o	Ω	ω	Θ
$\log_{10} n^2$	$\log_2 n$					
$n \log^3 n$	$n^{5/4}$					
$n^{\log(n)}$	$\sqrt{n}$					
$(\log n) n^k$	c^n					
$c^{\lg n}$	$n^{\lg c}$					
$n!$	2^{n^2}					

   (b) (5 pts) State the definition of Θ.

2. (20 pts total) Professor Turing drives an automobile from Salem to San Francisco along Interstate 5. His car's gas tank, when full, holds enough gas to travel n miles, and his map gives distances between gas stations on his route. The professor wishes to make as few gas stops as possible along the way.

   (a) (5 pts) Give an efficient method by which Professor Turing can determine at which gas stations he should stop.

   (b) (15 pts) Prove that your algorithm gives the optimal result.

3. (15 pts total) Huffman

   (a) (10 pts) Generate a Huffman code for the characters with the (character, frequency):
   (‘ ’, 6), (‘a’, 14), (‘b’, 2), (‘d’, 5), (‘h’, 1), (‘i’, 15), (‘t’,7).
   Be sure to show your work!

   (b) (5 pts) Write the encoding for the word “bad”

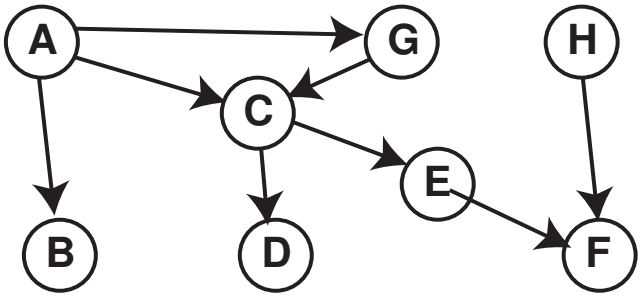
4. (15 pts total) For the following algorithms, *describe* (don't just give the name of) the problem the algorithm solves.

   (a) (5 pts) Dijkstra's Algorithm

   (b) (5 pts) Kruskal's Algorithm

   (c) (5 pts) Union-Find.

5. (15 pts total) Graphs:



(a) (5 pts) Show the adjacency list for the above graph. In places where more than one ordering of nodes is allowed, *always choose an alphabetical ordering.*

(b) (10 pts) Topologically sort this graph. Show your work and briefly explain how the algorithm works. *Note, when faced with a choice, use an alphabetical ordering.*

6. (10 pts) Design an algorithm to find a vertex in a connected undirected graph whose removal *does not* disconnect the graph. Argue why it is correct. The algorithm should run in linear time. As a consequence, argue that every connected graph contains such a vertex.

7. (10 pts Extra Credit) Find two functions $f(n)$ and $g(n)$, both monotonically increasing such that $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$. Prove your result.