Name: ____________________

CSE 421: Design and Analysis of Algorithms

Midterm Exam

February 11, 2000

Instructions: Give your answers in the spaces provided on these sheets. You need not justify your answer unless you are explicitly asked to. Feel free to describe an algorithm in English if that is more convenient than writing pseudocode.
Time limit: 50 Minutes.

1	/ 25
2	/ 25
3	/ 25
4	/ 25
Tot	/100

1. (25 points) Circle the most precise classification applicable to each of the following pairs of functions:

   A $f(n) = O(g(n))$, but it is not true that $f(n) = \Omega(g(n))$.

   B $f(n) = \Omega(g(n))$, but it is not true that $f(n) = O(g(n))$.

   C $f(n) = \Theta(g(n))$.

   D $f(n) = o(g(n))$.

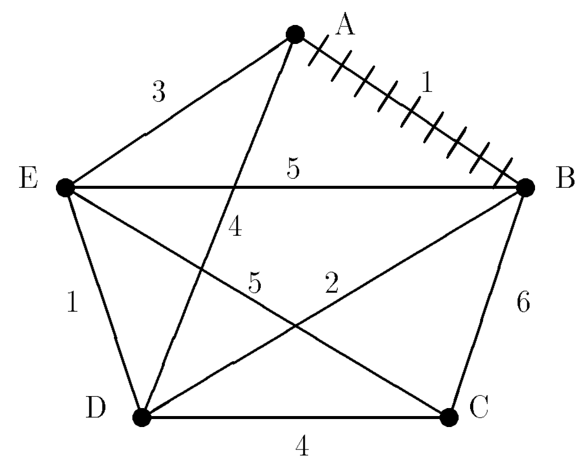
	$f(n)$	$g(n)$					
a	$500n - 1729n^2 + n^3$	n^3	....	A	B	C	D
b	$500n - 1729n^2 + n^3$	n^4	....	A	B	C	D
c	$n/\log n$	n	....	A	B	C	D
d	$n/\log n$	$\sqrt{n}$	....	A	B	C	D
e	n^{1000}	2^n	....	A	B	C	D
f	n^{1000}	1.01^n	....	A	B	C	D
g	$\sum_{i=1}^{n}(i^2+5)$	n^2	....	A	B	C	D
h	$\sum_{i=1}^{n}(i^2+5)$	n^3	....	A	B	C	D

2. (25 points)

   (a) Simulate a minimum spanning tree algorithm on the graph shown below. Use either Kruskal's algorithm (presented in lecture) or Prim's algorithm (presented in section 7.6). *Say which one you're using!* List, in order, the edges considered, their weight, and the action taken by the algorithm, namely "discarded", or "selected to be in tree". Highlight the selected edges, say, by drawing a squiggly line along each selected edge in the figure. The first edge is done for you below.

   Algorithm: ________________________

Edge	Weight	Action
1. AB	1	Selected
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		



(b) The graph above has only one minimum spanning tree; call it T. If we add the edge AC to the graph, the new graph may have the same minimum spanning tree, a new minimum spanning tree, or both.

i. Explain how you can tell which case applies by examining just the weighted graph $T \cup \{AC\}$.

ii. Which of these cases applies if the edge AC has weight 5?

iii. Weight 4?

iv. Weight 3?

3. (25 points) In the following table, we have used the dynamic programming algorithm to begin computing the optimal string alignment(s) of the two strings "**y = z = y + a ;**" and "**y = y + b ;**". As in the example used in homework, assume that aligning two identical letters gives a score of +2, whereas aligning a letter with a mismatched letter or a gap gives a score of −1.

	j	0	1	2	3	4	5	6	7	8
i			y	$=$	z	$=$	y	$+$	a	;
0		0	−1	−2	−3	−4	−5	−6	−7	−8
1	y	−1	2	1	0	−1	−2	−3	−4	−5
2	$=$	−2	1	4	3	2	1	0	−1	−2
3	y	−3	0	3	3	2	4	3	2	1
4	$+$	−4	−1	2	2					
5	b	−5	−2	1	1					
6	;	−6	−3	0	0					

(a) Fill in the remainder of this table.

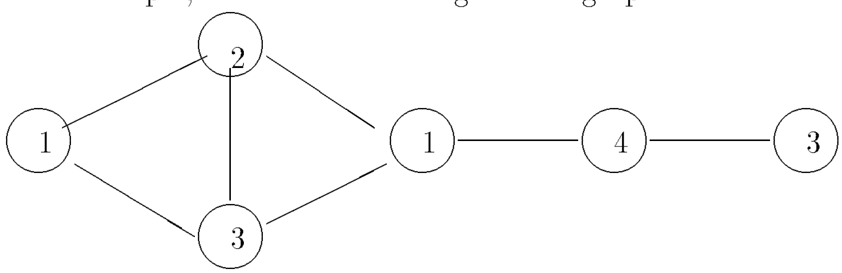
(b) There are two distinct maximal scoring alignments of these two strings, which you will find by tracing paths through this table. Starting at the lower right corner of the table, draw an arrow from each entry to *each* of its 3 neighbors that could have been responsible for its value. Some entries will have more than one arrow drawn from them. By only continuing this process for entries that have an arrow drawn *to* them, you will save yourself the time of filling in a lot of unnecessary arrows.

(c) From your paths that you have just identified in part (3b), give the two distinct alignments that have the highest score.

4. (25 points) A k-*coloring* of an undirected graph $G = (V, E)$ is an assignment of one of k colors to each of the vertices so that no two adjacent vertices have the same color. (Nonadjacent vertices *may* have the same color.)

   "Brooks' Theorem" states that every undirected graph G can be $(\Delta + 1)$-colored, where Δ is the maximum degree of any vertex in G.

   For example, here's a coloring of a graph of max degree 3 using 4 colors.



   (a) Prove Brooks' Theorem. (Hint: induction on $n = |V|$.)

   (b) Design an efficient algorithm to find such a coloring. Present your algorithm using the inductive framework stressed in Chapter 5 of the text. Briefly argue its correctness, if it's not immediately clear from 4a. Estimate its running time ($O(n + m)$ should be possible, where $m = |E|$).