TEST 3

Davis
CS222

Name:
Pledge:

Show all work; unjustified answers may receive less than full credit.

(10pts.) **1.** Write pseudo code for a recursive algorithm that inputs the the number of times compounding per year, and outputs the amount of money at the end of a year. Assume that you start with \$1000 and the annual interest rate is 12%. Compute the amount of money you will have if the interest is compounded 4 times in a year.

1. Compound(n)

2. int $= \frac{.12}{n}$

3. Recursivepart(n)

4. If $n = 0$ then $Total = 1000$

5. $Total = (1 + int)Recursivepart(n-1)$

6. end Compound(n)

(15pts.) **2.** Solve the recurrence relation $a_n = a_{n-1} + 12a_{n-2}$ with initial conditions $a_0 = 1, a_1 = 1$. Verify that your formula works for $n = 3$.

Assuming that $a_n = \alpha^n$, we get $\alpha^n = \alpha^{n-1} + 12\alpha^{n-2}$, so $\alpha^2 - \alpha - 12 = 0$. This factors into $(\alpha-4)(\alpha+3) = 0$, so $\alpha = 4$ or $\alpha = -3$. The general solution is $C_1 4^n + C_2(-3)^n$, and we can use the initial conditions to solve for C_1 and C_2.

$1 = C_1 + C_2; 1 = 4C_1 - 3C_2$. Multiply the first equation by 3 and add the equations, yielding $7C_1 = 4$, so $C_1 = \frac{4}{7}$, implying that $C_2 = \frac{3}{7}$.

Verifying the $n = 3$ case, the recurrence relation yields $a_3 = a_2 + 12a_1 = 13 + 12(1) = 25$. The formula yields $\frac{4}{7}4^3 + \frac{3}{7}(-3)^3 = \frac{256-81}{7} = \frac{175}{7} = 25$.

(15pts.) **3.** What conditions on r and s guarantee that the complete bipartite graph on $r + s$ vertices has an eulerian cycle? Find, if possible, an eulerian cycle in $K_{2,6}, K_{3,5}$, and $K_{4,5}$ (number the edges so I know which order you have drawn the picture).

r and s must both be even since all vertices must have even degree (the degree in the complete bipartite graph is either r or s). The eulerian cycle is easy to sketch out in the $K_{2,6}$ case, and the other two do not have an eulerian cycle.

(15pts.) **4.** Show by induction that there is a Hamiltonian cycle in the hypercube for $n \geq 2$ (the n-hypercube has vertices that are n-tuples of 0s and 1s, and two vertices have an edge between them if they differ in only one component).

See the argument on p. 288 in the book.

(10pts.) **5.** Construct a Huffman code from the following frequency table.

A	B	C	D	E	F	G	H
12	5	8	6	25	4	7	2

The first thing to do is to combine the two lowest frequency elements of the chart, namely F and H (for a total of 6). I will combine the next lowest element, B, with D at the next stage: you could combine B with FH. I now combine FH with G, and then combine C with BD. We now combine A with FGH to get 25, and I will combine E with BCD to get BCDE. We get the following strings from these choices: A = 01; B = 1000; C = 101; D = 1001; E = 11; F = 0000; G = 001; H = 0001.

(15pts.) **6.** Dijkstra's algorithm is the following:

1. procedure dijkstra(w,a,z,L)

2. $L(a) := 0$

3. for all vertices $x \neq a$ do

4. $L(x) := \infty$

5. T:=set of all vertices

9. $T := T - \{v\}$

10. for each $x \in T$ adjacent to v do

11. $L(x) := \min\{L(x), L(v) + w(v,x)\}$

12. end

13. end dijkstra

Show how this algorithm can be used to trace the shortest path through the graph listed below in matrix form.

$$
\begin{array}{c}
\\ a \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ z
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} a & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & z \end{array} \\
\begin{pmatrix}
-1 & 1 & 3 & 5 & -1 & -1 & -1 & -1 \\
1 & -1 & -1 & -1 & 18 & 14 & -1 & -1 \\
3 & -1 & -1 & -1 & -1 & 10 & 6 & -1 \\
5 & -1 & -1 & -1 & -1 & -1 & 2 & -1 \\
-1 & 18 & -1 & -1 & -1 & -1 & -1 & 1 \\
-1 & 14 & 10 & -1 & -1 & -1 & -1 & 5 \\
-1 & -1 & 6 & 2 & -1 & -1 & -1 & 9 \\
-1 & -1 & -1 & -1 & 1 & 5 & 9 & -1
\end{pmatrix}
\end{array}
$$

In step 1, remove a and relabel v_1 to 1; v_2 to 3; and v_3 to 5. In step 2, remove v_1 since that is the minimum label, and relabel v_4 to $1 + 18 = 19$; v_5 to $1 + 14 = 15$. In step 3, remove v_2 and relabel v_5 to $3 + 10 = 13$ (we do this since $13 < 15$); v_6 to $3 + 6 = 9$. In step 3, remove v_3 and relabel v_6 to $5 + 2 = 7$ (since $7 < 9$). In step 4, remove v_6 and relabel z to $7 + 9 = 16$. In step 5, remove v_5 and don't change any labels. In step 6, remove z and don't change any labels. This ends the algorithm since z is not in T anymore.

(20pts.) **7.** Use the breadth-first and depth-first algorithms to get spanning trees for the graph listed below in matrix form. Use $s = v_1$ for both algorithms.

$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12}
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} & v_{12} \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0
\end{pmatrix}
\end{array}
$$

Breadth-first algorithm:

1. $S := v_1$

2. $V' := \{v_1\}$

3. $E' := \phi$

4. while true do

5. begin

6. for each $x \in S$, in order, do

7. for each $y \in V - V'$, in order, do

8. if (x, y) is an edge then

9. add edge (x, y) to E' and y to V'

10. if no edges were added then

11. return(T)

12. $S :=$ children of S ordered consistently with the original vertex ordering

13. end

Depth-first algorithm:

$w = v_0$
while true do
begin
while there is an edge (w, v) that when added to T does not create a cycle in T do
begin
choose the edge (w, v_k) with minimum k that when added to T does not create a cycle in T
add v_k to V'
$w := v_k$
end
if $w = v_1$ then
return(T)
$w :=$ parent of w in T
end