CS435 Algorithms Final Exam

February 2, 2006

Name _____

Answer True or False to each question in this section. Please write clearly. [2 points each]

1. True To use the greedy method, a problem should have the property that a series of locally-optimal choices lead to a global optimal configuration.
2. F The greedy approach should never be used on optimization problems.
3. F The *task scheduling* problem is an example of the divide-and-conquer method because tasks are sorted by their running time before assigning them to machines.
4. T The *fractional knapsack* problem can be solved using the greedy method by continued selection of the item with the highest benefit-to-weight ratio.
5. T Recurrence equations are used to evaluate the time-complexity of divide-and-conquer algorithms.
6. F The problem of multiplying big integers of size n has $O(n^{1.585})$ time-complexity.
7. T The dynamic programming technique is similar to the divide-and-conquer approach in the way that a problem is divided in smaller, independent sub-problems and the results merged together to form the solution.
8. T When the dynamic programming technique is applied to the *multiplying matrices* problem the time-complexity is reduced from exponential to linear.
9. T Dynamic programming algorithms have a running time that only depends on n.
10. T Two vertices that are adjacent are endpoints of the same edge.
11. T The sum of the degrees of all vertices in a graph G are equal to the number of edges.
12. F A spanning tree of a graph contains only some of the vertices of the graph.
13. T An adjacency list structure has similar performance to the edge list structure but also has performance improvements in methods such as incidentEdges(v) and areAdjacent(u,v).
14. T Depth-first search traversal of an undirected graph uses the backtracking technique to explore the vertices and edges of a graph.
15. F In breadth-first search, the edges are marked as either discovery or cross edges to indicate their role in the spanning tree.
16. T To test whether a graph is connected, a DFS traversal can be performed and if some of the vertices are not marked as discovered, the graph is not connected.
17. T In a computer network, reachability in a directed graph is computed to find out if a message can be routed from node v to node w.

18. F The transitive closure of a graph measures the density of the edges in the graph.
19. T A topological ordering of a digraph is useful in scheduling tasks that have constraints as represented in the graph.
20. F Single-source shortest path algorithms find all the paths between a vertex *v* and *w* in a weighted graph.

Multiple choice questions. Pick the best answer. [3 points each]

21. The depth-first search algorithm we studied uses c.
    a) edge reversal
    b) a min-heap
    c) recursion
    d) transitive closure
22. The d problem can be solved optimally with the greedy approach.
    a) sum-of-subsets
    b) traveling salesman
    c) big integer multiplication
    d) fractional knapsack
23. The iterative substitution method is a technique that depends on our ability to a that can be converted to the closed-form version of the recurrence equation.
    a) see a pattern
    b) multiply matrices
    c) draw a tree
    d) apply a formula
24. In the following recurrence relation, the step of merging sub-problem solutions is done c times at each level.

$$T(n) = \begin{cases} 5b & \text{if } n < 2 \\ 3T(n/2) + 4n & \text{if } n \geq 2 \end{cases}$$

    a) 2
    b) 3
    c) 4
    d) 5

25. The a _____ applies to a problem if the best solution always contains optimal solutions of all sub-problems.
    a) principle of optimality
    b) greedy method
    c) iterative method
    d) big O notation
26. Which data structure is preferred when we need space efficiency when representing a graph with 10,000 vertices and 200,000 edges and we also need fast response to the areAdjacent method?
    a) adjacency matrix
    b) adjacency list <-----
    c) edge list

27. Which of the following problems has not been proven to be intractable, but also does not have a polynomial-time solution?
    a) 0-1 knapsack problem <-----
    b) minimum spanning tree problem
    c) matrix multiplication problem
    d) searching problem
28. What is an application of topological ordering?
    a) sorting mail by the user's name <-----
    b) finding the path between a pair of vertices
    c) determining connectivity in a graph
    d) showing the inheritance hierarchy in Java interfaces
29. A weighted graph will only have one minimal spanning tree if _____.
    a) every edge has a different weight <-----
    b) every edge has the same weight
    c) every vertex connects to very other vertex
    d) every vertex is a separate component
30. A ____ is a sequence of vertices that have an edge between each vertex and its successor.
    a) cycle
    b) map
    c) component
    d) path <-----

31. Problems that are in NP can be _____ in polynomial time.

a) solved

b) verified <-----

c) analyzed

d) halted

32. Some dynamic programming algorithms that we studied have a worst-case time complexity that is ____ , where the running time depends on the magnitude of a number in the input.

a) exponential <-----

b) quadratic

c) differential

d) pseudo-polynomial

33. [6 points] Using the Master Method, find the closed-form expression of the following recurrence equations:

$$T(n) = \begin{cases} c & \text{if } n < d \\ aT(n/b) + f(n) & \text{if } n \geq d \end{cases}$$

1. if $f(n)$ is $O(n^{\log_b a - \varepsilon})$, then $T(n)$ is $\Theta(n^{\log_b a})$
2. if $f(n)$ is $\Theta(n^{\log_b a} \log^k n)$, then $T(n)$ is $\Theta(n^{\log_b a} \log^{k+1} n)$
3. if $f(n)$ is $\Omega(n^{\log_b a + \varepsilon})$, then $T(n)$ is $\Theta(f(n))$,
   provided $af(n/b) \leq \delta f(n)$ for some $\delta < 1$.

a) $T(n) = T(n/3) + n \log n$

T(n) = O (nlogn)

b)

$T(n) = 8T(n/2) + n^2$

T(n) = O(n^2)

c) $T(n) = T(n/2) + 1$

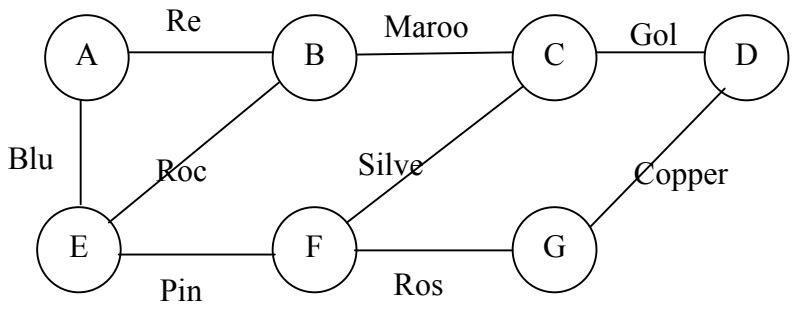
O(n)

34. [8 points] Solve the following problem using the 0-1 Knapsack algorithm. Show your calculations in a table format. What is the maximum benefit? Circle the final answer.

A machine designer is selecting parts to use in creating her newest creation. Her aim is to maximize benefit with the highest quality parts that fit within the space limits of her design. The volume is limited to 10 cc. The parts list is as follows (listed by value/volume):

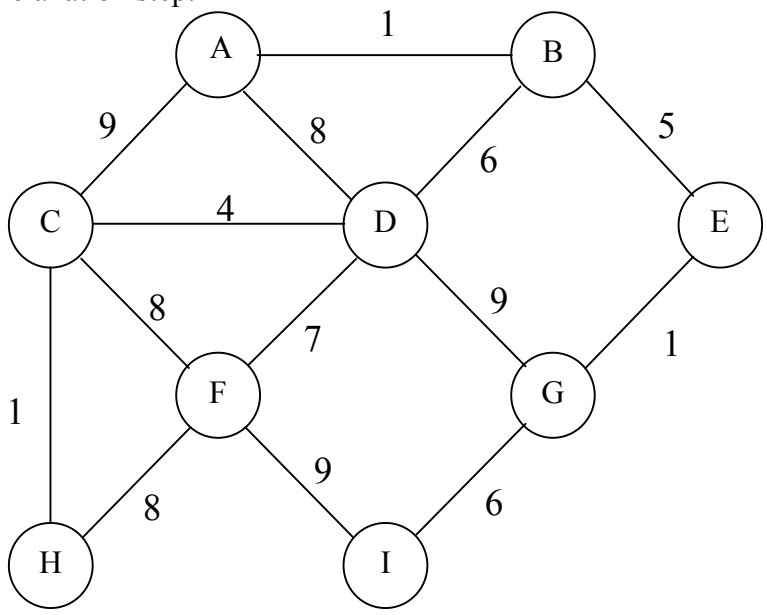
a: 9/3 b: 5/2 c: 12/3 d: 15/4 e: 8/3

35. [8 points] Draw an adjacency list representation of the following graph. Include all edges and vertices and draw arrows relating them as appropriate.



A = {B, C}
B = {D, E}
C = {F}
D = {G}
E = {H}
F = {I}
G = {}
H = {}
I = {}

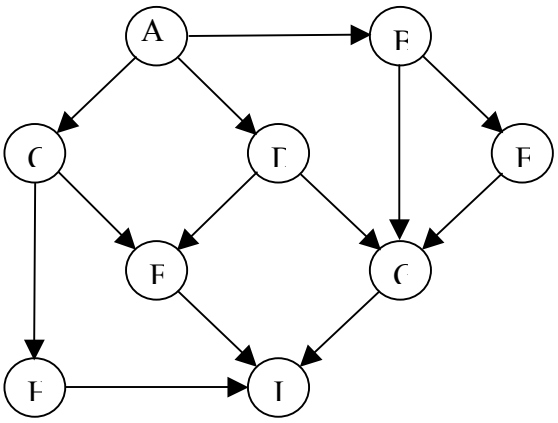
36. [8 points] Show the execution of Dijkstra's shortest path algorithm for the following graph starting from vertex A. Fill the table with the lengths of the best paths found so far for each vertex after each relaxation step.



Next Vertex	B	C	D	E	F	G	H	I	Edge Connecting Next Vertex to Cloud
A									

Vertex	B	C	D	E	F	G	H	I
Start	0	INF	INF	INF	INF	INF	INF	INF
A	0	1	6	5	1	2	6	9
C	1	0	5	4	1	3	7	10
D	6	5	0	6	2	6	9	15
E	5	4	6	0	3	5	8	13
F	1	1	2	3	0	4	7	12
G	2	3	6	5	4	0	9	14
H	6	7	9	8	7	9	0	15
I	9	10	15	13	12	14	15	0

37. [10 points] Bonus question: The *longest path* in a directed acyclic graph is a directed path with the maximum number of edges. For example, in the following graph the path ABEGI is the longest path. Describe in pseudo-code or Java, an O(n + m) time algorithm that computes a longest path in a directed acyclic graph. The algorithm should return the sequence of vertices on the path in the order they appear. Solve the general case, not just the one in the drawing. Analyze the run-time of your algorithm.



Note: You can call methods in the Graph ADT.

Algorithm FindLongestPath(G)

Input: An acyclic digraph G with n vertices

Output: A list of vertices that describe the longest path in G.

Algorithm FindLongestPath(G)Input: An acyclic digraph G wit