CS 170, Spring 1999
Midterm Exam II
Professor Christos Papadimitriou

1. **(2 points each question) True or false?** Circle the correct answer. No points will be subtracted for incorrect answers so guess all you want.

T F In an undirected graph, the shortest path between two nodes always lies on some minimum spanning tree.

T F If all edges of a graph have different weights the minimum spanning tree is unique.

T F If all edges of a graph have different weights the second-best minimum spanning tree is unique.

T F Prim's algorithm for computing minimum spanning trees runs in $O(|V^2|)$ steps.

T F In a depth first search on a directed acyclic graph, the vertex with the highest postorder number is necessarily a source.

T F In Huffman coding, the item with the second-lowest probability is always at the leaf that is furthest from the root.

T F In Huffman coding, the item with the highest probability is always at the leaf that is closest to the root.

T F In Huffman coding, the item with the highest probability is always at a leaf that is the child of the root.

T F If a set of Horn clauses contains no clause with a single literal, then it is satisfiable.

T F A set of Horn clauses can be tested for satisfiability in linear time.

T F In a depth-first search of a directed graph, it is possible to have an edge between two vertices *u* and *v* with previsit and postvisit numbers: (10, 40) for *u* and (30,50) for *v*.

T F In a depth-first search of an undirected graph, it is possible to have an edge between two vertices *u* and *v* with previsit and postvisit numbers: (5, 20) for *u* and (30, 50) for *v*.

T F In a depth-first search of a directed graph, it is possible to have an edge between two vertices *u* and *v* with previsit and postvisit numbers: (5, 20) for *u* and (30, 50) for *v*.

T F In a depth-first search of a directed graph, if *u* has pre and postvisit numbers (10, 20) and *v*, (15, 18), then (*v*,*u*) must be a back edge.

- **(15 points total)** In the weighted graph shown below,

  – **(4 points)** In running Dijkstra's algorithm for shortest distances from node A, which nodes will be inserted into the heap more than once? How many times?

  – **(4 points)** In running Prim's algorithm for minimum spanning tree from node A, what is the sequence of edges added to the minimum spanning tree?

  – **(7 points)** In running Kruskal's algorithm for minimum spanning tree, what is the union-find tree (*with* path compression) after the end of the algorithm? Without compression? (Recall that Kruskal's algorithm examines all edges of the graph.)

- **(15 points total)** The police department in the city of Computopia has made all streets one-way. The mayor contends that there is still a way to drive legally from any intersection in the city to any other intersection, but the opposition is not convinced. Furthermore, the city elections are coming up soon, and there is just enough time to run a *linear-time algorithm*.

  – Formulate this problem as a graph-theoretic problem, and explain why this problem can indeed be solved by a linear-time algorithm.

  – Suppose now that the mayor needs to show that, even though her original claim was false, something weaker does hold: If you start navigating one-way streets from the townhall, there is always a way to drive legally back to the town hall. Formulate this weaker property as a graph-theoretic problem, and carefully show how this problem too can be solved by a linear-time algorithm.

- **(30 points total)** We are given a strongly connected directed graph $G = (V, E)$ with positive edge weights, and let $v_0 \in V$. We want to find the shortest paths between *all pairs of nodes*; however, we are only interested in paths that go through the particular node v_0.

  – **(20 points)** Give an $O(|V|^2)$ algorithm for solving this problem.
  – **(5 points)** Do you think that there is a faster algorithm for this problem? Briefly support your answer.
  – **(5 points)** Prove or give a counterexample: The shortest path between nodes a and b through node v_0 in G, where $v_0 \neq a,b$ has no repeated edges.

- **Dynamic Programming (25 points total)** We are given twp strings x and y of length m and n, respectively. We are asked to find the *edit distance* between these two strings, that is, the minimum number of operations needed to transform x to y, when these kinds of operations are allowed: (i) insert a character in any position (ii) change one character into another (iii) delete *a whole consecutive block of characters of x).* Each of these three operations counts as one step. Find a dynamic programming algorithm that solves this problem, as follows:

  Define, for $i = 0,...,m$ and $j = 0,...,n$ $ED[i,j]$ to be the edit distance between the i characters of x and the first j characters of y.

  (Extra Credit:) **(10 points)** Can you devise an $O(m \cdot n)$ algorithm for this problem?