Quiz 2 – Practice Problems

1. Let G = (V, E) be a DAG, where each edge is annotated with some positive length. Let s be a source vertex in G. Suppose we run Dijkstra's algorithm to compute the distance from s to each vertex $v \in V$, and then order the vertices in increasing order of their distance from s. Are we guaranteed that this is a valid topological sort of G?

2. Suppose we have an alphabet with only five letters A, B, C, D, E which occur with the following frequencies.

Letter	A	B	C	D	E
frequency	0.35	0.12	0.18	0.05	0.30

Use Huffman coding to find the optimal prefix-free variable-length binary encoding of the alphabet.

(a) Draw a binary tree that represents the optimal encoding.

(b) Fill in the table below with the binary encoding of each letter.

Letter	encoding
A	
B	
C	
D	
E	

3. Given a set $\{x_1 \le x_2 \le \ldots \le x_n\}$ of points on the real line, determine the smallest set of unit-length closed intervals (e.g. the interval [1.25,2.25] includes all x_i such that $\{1.25 \le x_i \le 2.25\}$) that contains all of the points. Give the most efficient algorithm you can to solve this problem, prove it is correct and analyze the time complexity.

4. Let T be a complete binary tree with n vertices. Finding a path from the root of T to a given vertex v in T using breadth-first search takes

a) O(n)
b) O(lgn)
c) O(nlgn)
d) O(logn)
e) O(n^2)

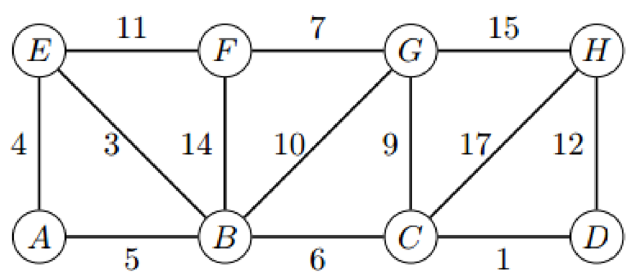
5. A Hamiltonian path in a graph G=(V,E) is a simple that includes every vertex in V. Design an algorithm to determine if a directed acyclic graph (DAG) G has a Hamiltonian path. Your algorithm should run in O(V+E). Provide a written description of your algorithm including why it works, pseudocode and an

explanation of the running time.

6. Consider a connected weighted directed graph G = (V, E, w). Define the *fatness* of a path P to be the maximum weight of any edge in P. Give an efficient algorithm that, given such a graph and two vertices u,v ∈V, finds the minimum possible fatness of a path from u to v in G.

7. Scheduling jobs intervals with penalties: For each $1 \leq i \leq n$ job j_i is given by two numbers d_i and, where d_i is the deadline and p_i is the penalty. The length of each job is equal to 1 minute. We want to schedule all jobs, but only one job can run at any given time. If job i does not complete on or before its deadline d, we should pay its penalty. Design a greedy algorithm to find a schedule which minimizes the sum of penalties.

8. Consider the weighted graph below:



Demonstrate Prim's algorithm starting from vertex A. Write the edges in the order they were added to the minimum spanning tree.