The Hong Kong University of Science & Technology

Department of Computer Science

COMP 271, Section 2: Design and Analysis of Algorithms

mid-term examination

Date: 12:00 - 2:00 p.m., 1 November, 1998.

Venue: LTD

Student name: ______________________________

Student ID: ______________________________

e_mail: ______________________________

Question	*Grade*
1	/12 pts
2	/12 pts
3	/12 pts
4	/12 pts
Total	/48 Pts

Problem 1:

Answer the following short questions.

(a) Discuss briefly why a correct algorithm is not always desirable.

(b) In the class, we talked about the Max-Dominance Problem. Given a set of points $P = \{p_1, p_2, \ldots, p_n\}$, each represented by its x and y coordinates, we want to find a set of maximal points of P. A maximal point p_i has the property that it is not dominated by any other point in P.

Argue briefly why if we output the maximal points in ascending order with respect to x, then we will have the y coordinates be descending with respect to y.

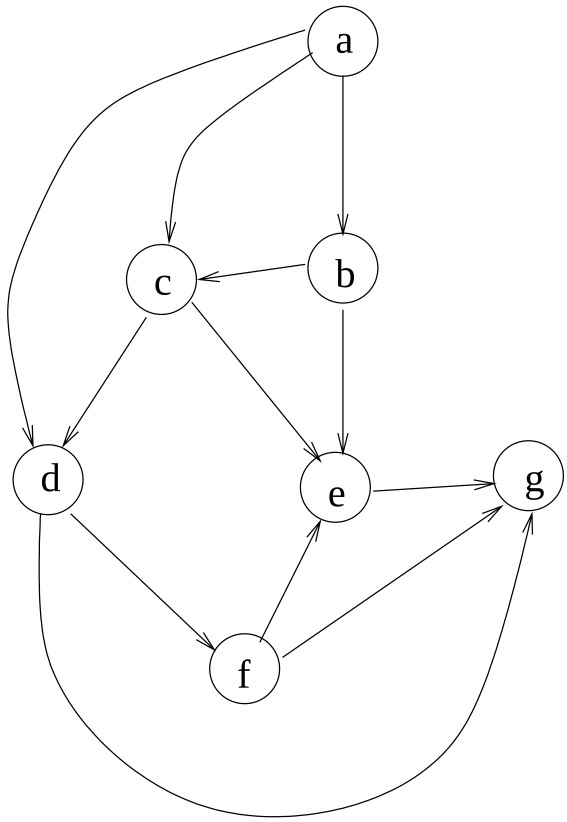
(c) Give one name of any *linear time* sorting algorithm. Discuss briefly the assumption for input numbers.

(d) Discuss how one real-life problem can be modeled as a graph problem.

Problem 2:

(a) Consider the following digraph $G = (V, E)$. Traverse the following digraph using Depth-First search starting from vertex a. You must use the adjacent list representation given below. Show the tree structure (both the edge classification and timestamps) produced during the search process.

$a :\rightarrow d, c, b$
$b :\rightarrow c, e$
$c :\rightarrow d, e$
$d :\rightarrow f, g$
$e :\rightarrow g$
$f :\rightarrow g, e$
$g :\rightarrow \phi$ (empty)



(b) What is *topological sort* of a directed acyclic graph ? What is the topological sort of the graph in part (a) ?

(c) Mr. Careless is a student of the class COMP 271. He claims that he can produce a topological sort of any directed graph. Discuss briefly if he is correct or he is wrong.

Problem 3:

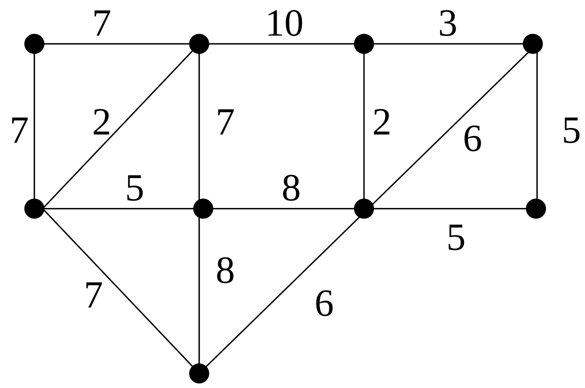
This question is about **minimum spanning tree** problem. Given an connected, undirected graph $G = (V, E)$, a spanning tree is an acyclic subset of edges $T \subseteq E$ that connects all the vertices together. Assuming that each edge (u, v) of G has a numeric weight or cost, $w(u, v)$, we define the cost of spanning tree T to be the sum of edges in the spanning tree

$$w(T) = \sum_{(u,v)\in T} w(u, v).$$

A minimum spanning tree (MST) is a spanning tree of minimum weight.

(a) In class we studied the **Kruskal's algorithm** to solve the MST problem. Write down the pseudo code of Kruskal's algorithm. Discuss briefly the idea of Kruskal's algorithm.

(b) Given the following graph $G = (V, E)$. Run Kruskal's algorithm on the graph. You only need to show the final minimum spanning tree, and compute the corresponding weight of the tree.



(c) The tricky part of Kruskal's algorithm is how to detect whether the addition of an edge will create a cycle. One way to solve the problem is to use the disjoint set UNION-FIND data structure. This data structure supports three operations:

Create-Set: Create a set containing a single item.

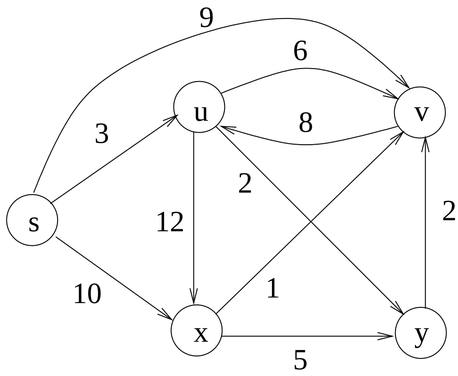
Find-Set: Find the set that contains a given item.

Union: Merge two sets into a common set.

Discuss how UNION-FIND data structure can help us to detect whether the addition of an edge will create a cycle.

Problem 4:

(a) You are given the following digraph $G = (V, E)$. Start with source s, trace the actions of Dijkstra's algorithm. Find the shortest paths from source s to all the other vertices. In each step, you need to show the d value of each vertex.



(b) Let $G = (V, E)$ be a weighted, directed graph such that the weights of the edges are nonnegative and are taken from the set $\{0, 1, 2, \ldots, W - 1\}$. Describe how we can use Dijkstra's algorithm to compute the shortest paths from a given source vertex s to all the other vertices in $O(WV + E)$ time. Argue briefly the running time of the algorithm.

(You need not to argue the correctness of the algorithm.)