Part II: Analysis and Design of Algorithms

*Answer as much as you can. You can attempt parts of a question. Use pseudo-code or code to describe required algorithms. Maximum grade = **50 pts.***

Question 1 [**10 Marks**] In the *online* median selection problem, a set of *n* integers are input *one by one* and the median is to be computed *n* times once after each integer is input. Describe an *O(n log n)* algorithm for solving the online median selection problem. Prove its running time.

Question 2 [**8 Marks**] Choosing a suitable data structure can improve the running time. **For each** of the following two algorithms, name the data structure that improved the running time and state the improved running time. (1) Dijkstra's shortest-path algorithm. (2) Kruskal's minimum spanning-tree algorithm.

Question 3 [**20 Marks**]
a. **[10 marks]** Describe an efficient algorithm to multiply two *n*-digit integers. What is the running time of your algorithm? Use the Master method to justify the running time.
b. [**10 Marks**] Describe a dynamic-programming algorithm to solve the 0–1 knapsack problem. The running time is O(nW) time, where *n* is number of items and *W* is the maximum weight of items that the thief can put in his knapsack. Is this algorithm a polynomial-time algorithm? Why?

Question 4 [**20 Marks]** Single-source shortest-paths algorithms include Breadth-First Search, Dijkstra's, and Bellman-Ford.
a. **[3 marks]** State the running time of each of the three algorithms.
b. **[2 marks]** Prove that Dijkstra's algorithm does not work correctly on graphs with **negative edge-weights**.
c. **[3 marks]** Explain why Bellman-Ford algorithm does not work correctly on a graph with **negative cycles**.
d. **[3 marks]** Explain why shortest-paths algorithms would not work correctly on a graph that is **not** connected?
e. **[4 marks]** Describe an algorithm to check graph connectivity in undirected graphs. What is its running time?
f. **[2 marks]** Describe how to compute shortest paths between **all the pairs** of the graph vertices.
g. [**3 marks**] Describe a methodology to **select** the best algorithm to compute shortest paths between **all-pairs** of graph vertices. (*Hint*: select an algorithm based on edge count, vertex count, and values of edge weights.)

Question 5 [**12 Marks**] Answer the following about the Minimum Spanning Tree (MST) problem.
a. **[2 marks]** Define the MST problem by specifying its inputs and outputs.
b. **[5 marks]** Describe an efficient algorithm for solving the MST problem.
c. **[5 marks]** Prove that your algorithm in (b) is correct (i.e., it computes a spanning tree with minimum cost). What is its running time?

Question 6 [7 marks] NP-completeness
a. ***[2 marks]*** Which (**if any**) of the following two figures is correct and why?
b. ***[3 marks]*** Sketch briefly how to prove that a problem is an NP-complete problem (problem reduction).
c. ***[2 marks]*** List four strategies to efficiently handle NP-complete problems.