

All discussions about the “idea of how to solve this question” or “whether this solution is correct” prohibited. You can only ask clarification questions about the problems but must not discuss any idea regarding solving the problem. No handwritten answers will be graded. You must type all your answer text descriptions in a text editor (this is because handwritten answers are difficult to read and check for plagiarism). You can only draw images by hand and add them to your answer. If you are unable to type for special reasons then discuss that with your instructor.

Marking scheme: You may assume the subject material refers to algorithms: general guidelines for the grading system for undergraduate students: <https://students.usask.ca/academics/grading/grading-system.php>GradingSystem.

Q1. Assume that a problem A cannot be solved in $O(n^2)$ time. However, we can transform A into a problem B in $O(n^2 \log n)$ time, and then solve B , and finally transform the solution of B in $O(n)$ time into a solution for A .

Prove or Disprove: The above approach shows that B cannot be solved asymptotically less than $O(n^2)$ time.

Q2. Prove that Minimum Vertex Cover problem is NP-hard by reducing the NP-hard problem Maximum Independent Set.

Q3. Since finding a minimum vertex cover in a graph is known to be NP-hard, we want to find a vertex cover S that is not too large than a minimum vertex cover. We propose the following algorithm.

Step 1. Initialize S with an empty set.

Step 2. While there is an edge in G , randomly choose an edge (a, b) and insert a and b into S . Then delete the vertices a and b , and all the edges incident to a and b .

Step 3. Return S

Prove that the size of the set S returned by the algorithm can be at most twice the size of a minimum vertex cover of G .

Q4. You are give a social network of n students, where two students are connected if and only if they are friends. Consider the problem of finding a smallest set S of students from the network such that any student in the network is either in S , or a friend of someone who belongs to S .

Either give a polynomial-time algorithm for the problem, or show that the problem is NP-hard by reducing the Minimum Vertex Cover problem.