University of Colombo School of Computing
SCS 1308 - Foundations of Algorithms
Take-home 04

Instructions

- **Try the following questions and upload your answer script as a zip file to the given link in the UGVLE on/before 05th January 2025 at 6 pm.**
- **Note: Rename your zip file with your index number and name. (i.e: indexNo_Name.zip).**

01. We are now going to reduce the Minimum Vertex Cover problem to the Clique Problem. Given a graph G(V,E), the complement of G is the graph $\overline{G}(V,\overline{E})$ that has the same set of vertices V, but if there is an edge $(u,v)\in E$, then there is no edge $(u,v)\in\overline{E}$, and vice versa.

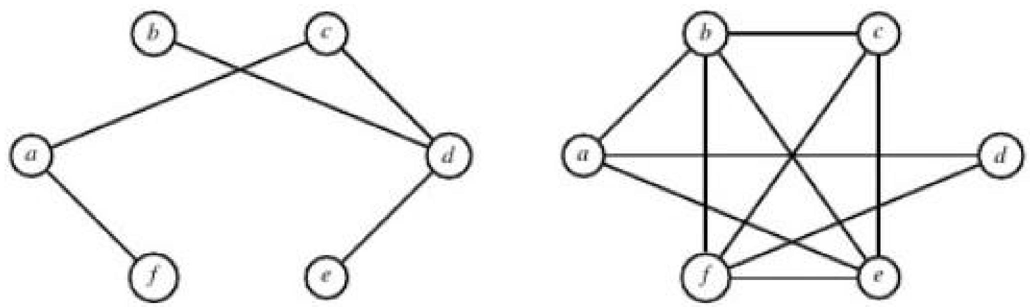


Figure 1: A graph G (on the left) and its complement $\overline{G}$ (on the right).

(a). Show that the decision version of the clique problem is in NP.

(b). Find a vertex cover of size less than or equal to 2 in G.

(c). Show that the complement of the vertex cover that you found in the previous part is a clique of size 6−2=4 in $\overline{G}$.

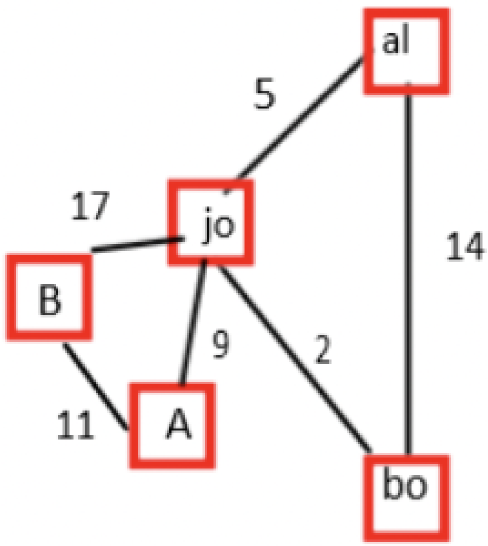
(d). Explain how you can reduce the minimum vertex cover problem to the clique problem.

(e). Can you also reduce it in the other direction?

02. Carefully prove that problem 1 is NPHard (i.e., can be solved in polynomial time if and only if P=NP) or show that it has a polynomial solution by giving the algorithm in pseudocode.

To prove it is NPHard, do a reduction of HAM. Use words such as "if" and "so" and "therefore" and "implies" when explaining your logic.

a). Given a connected undirected graph with V vertices and positive weights on its edges. Determine the cost of the cheapest path that visits at least V−2 different vertices. (Example: answer is 7: al → jo → bo is the path.)



03. All of the NP-Complete problems we looked at in class were decision problems, in that we were trying to decide if a condition was true or not (e.g. “Does the graph have a clique of size k?”). However, for real problems we often would want to solve the max/min problem (often called the optimization problem), for example, “What is the largest clique in the graph?”. I claim that in most situations if you can solve the decision problem in polynomial time then you can solve the optimization problem in polynomial time. Prove that this is true for the **CLIQUE** problem.

04. Assume only the basic knowledge of NP-complete problems; namely, Satisfiability problem, Hamiltonian Cycle problem, and Subset-Sum problem are NP-complete. And assume P≠NP.

a). Determine whether each of the following two problems (A) and (B) is in P or is NP-hard.

b). If it is in P, describe an algorithm briefly, and state the time complexity.
If it is NP-complete, prove your answer.

(A) The set-partition problem. The set-partition problem takes as input a set S of numbers. The question is whether the numbers can be partitioned into two sets A and B=S−A such that the sum of the numbers in set A is equal to the sum of the numbers in set B:

$$\sum_{x \in A} x = \sum_{x \in B} x$$

(B) The 3-partition problem. The 3-partition problem takes as input a set S of numbers. The question is whether the numbers can be partitioned into three sets A,B, and C, such that A∪B∪C=S, and the sum of the numbers in set A, the sum of the numbers in set B, and the sum of the numbers in set C are all equal:

$$\sum_{x \in A} x = \sum_{x \in B} x = \sum_{x \in C} x$$