You are allowed to discuss with others but are not allowed to use any references other than the course notes and the reference books. Please list your collaborators for each question. You must write your own solutions. See the course outline for the homework policy.

This homework accounts for 8% of your total grade.

Unless otherwise stated, giving an algorithm means providing a high level description, pseudocode (if needed to clarify the algorithm), a correctness argument, and runtime analysis.

Important:

To make things easier for the TAs your NP-completeness proofs should have the following format, with these steps clearly labelled:

(i) a proof that the given problem lies in NP

(ii) which known NP-complete problem you will use for your transformations, and a statement (using $\leq_P$) about which problem you will reduce to which

(iii) your transformation, i.e., how to convert an input for one problem (be clear which one) to an input of the other problem

(iv) a proof of correctness

(v) a justification that your transformation takes polynomial time (in most cases, this can be brief).

For this assignment, "known NP-complete problems" you may use in your transformations (or reductions) include: (3-)SAT, Clique, Vertex Cover, Hamiltonian cycle/path (directed or undirected), and Subset Sum. Definitions of these problems can be found in the textbooks.

Unit costs are fine for runtime. Input size is measured in bits.

Every term some students get the direction of their reductions wrong and lose all marks. Please do your best to carefully choose the directions for your reductions!

Due date: December 5th at 11:59pm

Problem 1 (25 Points) - Seating Plan

You are in charge of proctoring an exam for n students. Because all rooms are booked, the exam is taking place in a hallway in a quiet part of the building in which n individual desks have been placed in one long row. Protocol dictates that you should avoid seating two students beside each other if one of the students knows the other student.

Specifically, the Seating-Plan problem is to find a permutation $[i_1, i_2, \ldots, i_n]$ of the first n integers such that student i_j doesn't know the student directly to the left of them ($j > 1$) nor directly to the right of them ($j < n$). For student i you are given a set $S_i \subseteq \{1, 2, \ldots, n\} \setminus \{i\}$ of students that student i knows, $1 \leq i \leq n$. (You may assume that if student i knows student j, then student j knows student i.) To be clear, the *output* is a *permutation*.

1. [15 marks] Give a polynomial time Turing reduction from Seating-Plan to Ham-Path, where Ham-Path is the problem of finding a simple path in an undirected graph that visits each vertex. (Note: the question asks for a Turing reduction because transformations are only used for decision problems, and here the problems have more general outputs. But your reduction will probably have many of the characteristics of a transformation.) Prove your reduction is correct.
2. [5 marks] True or false. Seating-Plan is in NP. Briefly justify your answer.
3. [5 marks] True or false. If you gave a correct polytime Turing reduction from Ham-Path to Seating-Plan you would have shown Seating-Plan is NP-hard. Briefly justify your answer.

PROBLEM 2 (25 Points) - Adaptor Chaining

Your algorithms professor just shared a spicy meme:

Inspired by this meme, you have gathered cable adaptors of various kinds from all over your house. You want to plug them into each other, for fun.[1]

The adaptor chaining problem

Input: adaptors $a_1, ..., a_n$. Each adaptor a_i has two ends, each of which has a particular *type*. Let l_i and r_i denote the types of the left and right ends of adaptor a_i.

If two adaptors have ends with the same type, they can be connected to one another. (The directions left and right are not special. You can reverse the direction of an adapter if you like, in order to connect it to another adapter, assuming the two adaptors have ends with compatible types.)

For the following variants of the adaptor chaining problem, either prove that the problem is NP complete, or find a polytime solution to the problem.

(i) [12 marks] Suppose you want to make a large **ring** of adaptors that uses each **connector type exactly once**. In this variant, you still need two adaptors to have a connector type in common to connect them, but once you've connected two adaptors with a particular type, e.g., USB-C, your chain shouldn't contain any *other* adaptors with that type. Is it possible to connect (a subset of the) adaptors in a ring such that each connector type is used exactly once? **Output:** yes or no.

(ii) [13 marks] Can all n adaptors be connected together to form one long chain? **Output:** yes or no.

[1] Is this what professors imagine their students do for fun? Is this what professors do for fun? Only an oracle would know...

Problem 3 (25 Points) - NP-completeness

1. [12 marks] Prove the following variant of the st-Path problem is NP-Complete via a polynomial transformation from Directed-Hamiltonian-Path (see definition at the beginning of the assignment).

   Modified-st-Path

   Input: Edge-weighted directed graph $G = (V, E)$ where each edge $e \in E$ has an integer weight w_e (could be negative, positive or zero), two designated vertices s and t, and an integer k.

   Output: Does there exist a *simple* s-t path with total weight at most k? (YES or NO)

2. [13 marks] For this part, assume the following problem, Subset-Sum-Zero, is NP-Complete.

   Subset-Sum-Zero

   Input: n integers (can be negative) $w_1, \ldots, w_n$

   Output: Does there exist $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i = 0$ (YES or NO)

   Prove that the following variant is NP-Complete via polynomial transformation from Subset-Sum-Zero.

   Subset-Sum-One

   Input: n integers (can be negative) $w_1, \ldots, w_n$

   Output: Does there exist $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i = 1$? (YES or NO)

Problem 4 (25 Points) - Michel Alain Star

Chef Michel Alain is renowned for his culinary skills and his willingness to cater to his patrons' diverse tastes. To enhance the dining experience at his restaurant, he decides to offer a special menu where diners can evaluate his dishes. The menu presents a variety of potential dishes in areas where Chef Alain is open to suggestions. Each diner is given the opportunity to express their preference for each dish as either *"absolutely love"*, *"indifferent"*, or *"absolutely dislike"*. However, to obtain the true preference of the patrons, each diner is restricted to select *"absolutely love"* or *"absolutely dislike"* for a maximum of five dishes.

Chef Alain's primary goal is to delight his patrons. A patron is considered delighted if at least one of their top-choice dishes (those they *"absolutely love"*) is included in the final menu, or if one of their least favorite dishes (those they *"absolutely dislike"*) is excluded. The challenge for Chef Alain is to design the final menu so as to delight every single one of his patrons.

Output: a menu that delights all patrons, or "impossible"

Problem: either prove that Chef Alain's challenge is in NP-hard or provide a polytime solution to this problem.