ASSIGNMENT 4

DUE: Tuesday July 26, 11:59pm. LATE SUBMISSIONS NOT ACCEPTED!!!
DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read http://www.student.cs.uwaterloo.ca/~cs341 for general instructions and policies.

Problems. To be handed in.

To make things easier for the markers 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 reduction, and a statement (using $\leq_P$) about which problem you will reduce to which
(iii) your many-one reduction (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 reduction takes polynomial time (in most cases, this can be brief).
1. [10 marks] In the CLIQUE3 problem, we are given a graph \( G = (V, E) \) with maximum degree 3 and a positive integer \( k \); we must determine if \( G \) has a clique of size at least \( k \) or not.

(a) Prove that CLIQUE3 ∈ NP.

(b) Here’s a claimed proof that CLIQUE is NP-complete. Explain why the argument is incorrect.

We showed in part (a) that CLIQUE3 is in NP.
We know from lectures that CLIQUE is NP-complete.
We complete the proof by showing that CLIQUE3 \( \leq_p \) CLIQUE: Let \( F \) be the (trivial) algorithm that takes in a graph \( G \) with vertices of degree at most 3 and a parameter \( k \), and leaves both as-is. The algorithm \( F \) runs in polynomial time and gives a transformation from inputs of the CLIQUE3 problem to inputs of the CLIQUE problem, and the answer to these inputs is always identical. Therefore, this is a valid polynomial-time reduction and CLIQUE3 is NP-complete.

(c) In the VERTEXCOVER3 problem, we are given a graph \( G = (V, E) \) with maximum degree 3 and a positive integer \( k \); we must determine if \( G \) has a vertex cover of size at most \( k \) or not. It is known that VERTEXCOVER3 is NP-complete, and for this question we may use this fact without proof.

Here’s another claimed proof that CLIQUE3 is NP-complete. Explain why the argument is incorrect.

We already showed in part (a) that CLIQUE3 is in NP.
As stated in the question, we know that VERTEXCOVER3 is NP-complete.
We complete the proof by showing that VERTEXCOVER3 \( \leq_p \) CLIQUE3: Let \( F \) be the algorithm that transforms the input \( (G, k) \) into the input \( (\overline{G}, n - k) \) where \( \overline{G} \) is the complement of \( G \)—the graph on the same set \( V \) of vertices as \( G \) where for every pair of distinct vertices \( u, v \in V \), \( (u, v) \) is an edge in \( \overline{G} \) if and only if it is not an edge in \( G \).
The algorithm \( F \) has polynomial-time complexity. And there is a vertex cover of size \( \leq k \) in \( G \) if and only if there is a clique of size at least \( (n - k) \) in \( \overline{G} \). Therefore this is a valid polynomial-time reduction and CLIQUE3 is NP-complete.
2. [10 marks] **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.

(a) 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 many-one reductions 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 many-one reduction.) Prove your reduction is correct.

(b) True or false. Seating-Plan is in NP. Briefly justify your answer.

(c) True or false. If you gave a correct polytime reduction from Ham-Path to Seating-Plan you would have shown Seating-Plan is NP-complete. Briefly justify your answer.
3. [10 marks] Prove that the following decision problems are NP-complete.

(a) **EVEN_SPLIT**: Given an array $A[1..n]$ of $n$ integers, determine whether there is a set $I \subseteq \{1,2,\ldots,n\}$ for which

$$\sum_{i \in I} A[i] = \sum_{j \in \{1,2,\ldots,n\} \setminus I} A[j].$$

For the purpose of this question, you can assume that the following variant of Subset Sum is NP-complete: **Target Subset Sum** takes an array of integers and an integer $T$ as its input, and returns true if a subset of the input array sums to $T$, and false otherwise.

(b) **LOW DEGREE SPANNING TREE**: Given a graph $G = (V,E)$ and a positive integer $k$, determine whether there is a spanning tree $T$ of $G$ where each vertex has degree at most $k$ in $T$.

Part (b) deleted to make assignment shorter / save you time.
Challenge Questions. This is for fun and enrichment only. Solutions will NOT be marked.

1. What about the Seating Plan problem where you are seating students in a square grid (like in the PAC in pre-covid times)? If student $A$ knows student $B$, then $B$ should not be seated in front, behind, to the left, or to the right of $A$. Is this problem NP-complete?