ASSIGNMENT 8

DUE: Tuesday, April 5, 11:59 PM*

*The due date for this assignment was originally stated, in error, to be Wednesday, April 6. To account for any inconvenience the change in due date may have caused, you may submit this assignment up to 24 hours late with no penalty. However, no further late days are allowed.

DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

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

Exercises. The following exercises are for practice only. You may ask about them in office hours. Do not hand them in.

1. Prove the following directed version of Hamiltonian path problem is NP-complete.

   **DIRECTED-HAMILTONIAN-PATH**
   
   **Input:** A directed graph \( G \).
   
   **Output:** Does \( G \) contains a path that visits every vertex of \( G \) exactly once? (YES or NO)

2. Prove that the decision version of 0/1-Knapsack problem is NP-Complete.

Problems. For this assignment the “known NP-complete problems” you may use for your reductions are: 3-SAT, Clique, Independent Set, Vertex Cover, Hamiltonian cycle/path (both directed and undirected), Subset Sum. These decision problems were defined in Lectures 19, 20, 21.

To make things easier for the markers, your NP-completeness proofs MUST 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 (with \( \leq_P \)) 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] Hardness of Maximum 2-SAT.

   Consider a 2-SAT instance: there are \( n \) boolean variables \( x_1, x_2, \ldots x_n \) and a list of \( m \) clauses

   \[
   l_{11} \lor l_{12}, \ l_{21} \lor l_{22}, \ldots, \ l_{m1} \lor l_{m2},
   \]

   where each literal \( l_{ss} \) is \( x_s \) or \( \neg x_s \) for some variable \( x_s \). Note that there is no requirement that the clauses be pairwise distinct.

   The Max 2-SAT problem is to find an assignment to the \( n \) variables \( x_s \) that satisfies the maximum number of these \( m \) clauses.
(a) [2 marks] Define a decision version MAX 2-SAT-Dec of problem MAX 2-SAT.

(b) [6 marks] Give a poly-time many-one reduction from CLIQUE to MAX 2-SAT-Dec. You must provide a direct reduction from CLIQUE to MAX 2-SAT-Dec.

Note: For this part problem, you solution should consist of steps (iii), (iv) and (v) as specified above.

(c) [2 marks] Conclude from parts (a) and (b) that MAX 2-SAT-Dec is NP-complete.


The minimum weight connected subgraph problem takes as input

- an undirected graph $G = (V, E)$ with positive edge weights, together with
- a subset of vertices $S \subseteq V$,

and requires as output a subset of edges $F \subseteq E$ such that for any two vertices in $S$, there is a path in the subgraph $(V, F)$ between them. Moreover, the sum of the weights of the edges in $F$ should be minimal among all such subgraphs.

Give a polynomial time 2-approximation algorithm for the minimum weight connected subgraph.

Hint: Construct a complete graph $H$ on $S$ such that a minimum spanning tree in $H$ corresponds to a subgraph of $G$ with the desired property.

Challenge Questions. This is for fun and enrichment only. Do not hand it in.

1. Give a polynomial time algorithm for solving 2-SAT, which is SAT where every clause has exactly two literals.

2. Given a $8/7$-approximation for max-3-SAT where each clause's three literals involve different variables.