<u>Problem set 4</u>
Computational Complexity.

1. We saw in class that $2 - Sat \in co - NL$. Prove that $2 - Sat \in co - NL - complete$.
2. Show with complete proofs:
   (a) $CSAT \leq_L SAT$
   (b) $SAT \leq_L 3SAT$
3. (a) Suggest an algorithm for the Subset-Sum problem that works in time which is polynomial in n and in S where n is the size of the input and S is the target sum (Hint: use a binary array of size S).
   (b) Explain why this does not prove that $P = NP$ (in your answer you may want to refer to the proof of NP-hardness of Subset-Sum).
4. Let Dominating-Set *(DS)* be the following problem:
   $DS = \{(G, k) \mid$ There is a subset U of the vertices of G of size k, such that every vertex is either in U or has a neighbor in $U\}$

   Prove that $DS \in NPC$.
5. Prove: $NSPACE(n) = co - NSPACE(n)$.
6. We saw in class that an NP machine can be changed so that all guesses are made at the beginning. We next show that this is probably not the case for NL.

   Let us define the class NL^* to be the class of all languages L for which there is a TM with the following criteria:
   Input tape: Read only, move in both directions.
   Witness tape: Read only, move in both directions.
   Work tape: Read-Write, move in all directions.
   The machine itself is deterministic (the guesses are the value of the witness tape). The space complexity is the size of the work tape, and is bounded by $O(\log n)$. We say the machine accepts an input x if and only if there exits a setting for the witness tape, with which the machine returns TRUE.

   (a) Prove that $NP \subseteq NL^*$.
   (b) Conclude that if $P \neq NP$ then $NL \neq NL^*$.