Complexity Theory

1. Prove that P closed under the star operator ($*$). (Hint: Given a language $L \in P$ and input a, use dynamic programming to build up solutions for each substring of a to see if $a \in L^*$).

   Then, show that NP is also closed under the star operator ($*$).

2. Show that if $P = NP$, then every language $A \in P$, except $A = \emptyset$ and $A = \Sigma^*$ is *NP-Complete*.

3. *PSPACE* is the complexity class containing all problems that can be solved using a *Deterministic Turing Machine* that uses at most a polynomial amount of additional space (on the tape). Prove that $P \subseteq$ *PSPACE*. (*Hint: Think about a generic problem in P, and the DTM that decides it. Try to use more than a polynomial space with this machine given the allotted time.*)

4. Let *Double-Sat* (DS) be the following: $DS = \{\theta \mid \theta$ has at least two satisfying assignments$\}$. Show that *DS* is *NP-Complete*.

5. You are working for *Amaphon* (an online realtor that definitely does not exist) and you are given the following task from your boss: Your users have a profile and may or may not have an in-store balance (i.e., a sum of money that can be directly spent on items on *Amaphon*). For example, maybe one user has $\$31.47$ in their account right now. Your team wants to build a new feature in which customers can select a type of product (e.g., Holiday gifts or clothes or electronics) and the site will automatically suggest a set of items they can purchase for which the total price is exactly their current balance. This means the customer gets a bunch of cool products AND they get to clear out their balance to exactly 0 all at once.

   To clarify this problem a bit, consider the following: You are given as input the users balance B, and a list of n products in their chosen category along with the prices of those products. Only the prices really matter, so let's call this list $P = \{p_1, p_2, ...p_n\}$ Your task is to find any subset of these prices that totals exactly B.

   You suspect this problem might be *NP-Complete*. Prove it! For your reduction, use *3-SAT*.