Lab #13A

#1 Developing the Strategy

1. Review the proof that SAT is in PSPACE (Example 8.3). In a sentence or two, describe how the construction captures the exponential nature of the search space using a polynomial amount of memory.
2. Next prove that the following problems are in PSPACE.
   - $\mathsf{CLIQUE} = \{ (G, k) \mid G \text{ has a } k\text{-clique} \}$
   - $\mathsf{SUBSET} = \{ (S, t) \mid S \text{ is a set, } t \in \mathbb{N}, \text{ and there exists a subset } C \subseteq S \text{ such that } \sum_{x \in C} x = t. \}$
3. Now that you've had some experience thinking about how PSPACE algorithms work, let's generalize our constructions. At this point, you should observe that there seems to be a pattern for every NP problem and realizing it as a PSPACE algorithm. This pattern is similar to the verifier-to-NTM direction of the proof of Theorem 7.20 (pp. 294).

   Adapt this direction of the proof to formally prove that NP $\subseteq$ PSPACE.

   (Hint: the definition of NP says that you start with a verifier for the problem. You must then build a polynomial space algorithm for the problem using this verifier. What do you use your polynomial space to "remember?")

#2 Space Relationships

1. Consider a Turing machine with q states and g possible tape symbols. Write down an expression for the number of possible configurations of this Turing machine with a tape size of n. Explain your expression in a sentence or two.
2. Prove the following fact regarding halting Turing machines and configurations:

   Claim 1. *A Turing machine that halts does not repeat a configuration.*
3. Observe that if a language is in PSPACE then a Turing machine that recognizes it uses a polynomial amount of space. Use your results from the previous two parts to prove the following claim:

   Claim 2. *PSPACE $\subseteq$ EXPTIME.*

   (Hint: establish the number of configurations of such a Turing machine and use the previous problem's claim to establish an upper-bound on the number of steps the Turing machine.)