CS 254: Computability and Complexity	**Anonymous submission**
Problem Set #01	*November 16, 2019*

1. Show that subsetSum $\leq p$ Partition.

   Assume there is a TM for partition. Then we can build a machine for subsetSum M

   M = "on input $\langle$S, t$\rangle$where S is a set of binary numbers and t is an integer target.

   1. Let s be the sum of S. and S' = S $\cup$ s-2t.
   2. run $\text{TM}_{partition}$ on S'
   3. If $\text{TM}_{partition}$ on S' accepts, accept; else reject.

   Validation: If $\langle$S,t$\rangle\in$ subsetSum iff $\langle$S'$\rangle\in$ partition.

   Since we can assume partition works properly, we can properly solve subsetSum because if

   If there is a set in S that sums to t, the remaining set must sum to s-t.

   Since we unioned S with s-2t to result in S', S' must contain the integers for t, s-t, and s-2t.

   If we combine t with s-2t, we see that S' contains two sets of s-t.

   Therefore the partition TM accepts, and proves $\langle$S,t $\rangle\Longrightarrow\langle$S' $\rangle\in$ partition.

   If there exists a partition of S' into two sets that each sum to s-t then one of the sets contains the number s-2t.

   If we were to remove the number s-2t from the set, the set remaining would sum to t because s-2t + t = s-t.

   Therefore this fulfils that any $\langle$S' $\rangle\Longrightarrow\langle$S,t $\rangle\in$ subsetSum.

   Therefore we know $\langle$S,t $\rangle\in$ subsetSum $\Longleftrightarrow\langle$S' $\rangle\in$ partition.

   We can see that this solves our problem $\in$ NP:

   step 1 runs in polynomial time because we sum the numbers and create a new set.

   step 2 runs $\in$ NP since we know partition is $\in$ NP

   step 3 is O(1) as all we are doing is returning

   Therefore this solves our problem $\in$ NP.

   Thus: subsetSum $\leq p$ Partition