Terms to know

Tractable: a program is tractable if it can be finished in a "reasonable amount of time", which we take to mean anything faster than O(2**n).

Reduction: a mapping of one problem onto another. If you get the solution to one, you can get the solution to the other

P: polynomial. A function is "in P" if it is O(n**k) for some integer k.

NP: nondeterministic polynomial. If we non-deterministically, or randomly, guess an answer, we can verify the answer's correctness in polynomial time. A problem is "in NP" if it fits this description.

P = NP?: for all problems in NP, which we have O(2**n) solutions for, is there a polynomial time solution?

Case study: subsetSum

Given a set s of ints, return True if there exists a subset of s that sums to 0, and False otherwise.

Our goal is to prove that subsetSum is in NP.

Need to check every subset. We will show that this is O(2**n).

For example, let s = {2, -5, 3}. We want to enumerate every subset of s. We do this by making a table:

Membership			**Subset**
2	**-5**	**3**	
0	0	0	{}
0	0	1	{3}
0	1	0	{-5}
0	1	1	{-5, -3}
1	0	0	{2}
1	0	1	{2, 3}
1	1	0	{2, -5}
1	1	1	{2, -5, 3}

This table enumerates all of the subsets of s. It has 2**3 rows, and each row expresses one subset of s.

We can generalize this, so that for a set of size n, our table will have 2**n rows to enumerate 2**n subsets. So the number of subsets of a set is 2**n.

To check if a single subset sums to 0, it takes O(n) time, because the subset contains at most n elements.

So, our solution to subsetSum is O(n*2**n), which is exponential. However, if we randomly guess a subset, we can check to see if it sums to 0 in O(n) time, which is Polynomial. Therefore, subsetSum is in NP.