CS390 Week 5 3.2.2025

Problem 11.5.1

I shall show that using the Partition Problem as a black box, I can solve Subset Sum in polynomial time (which implies I can solve SAT in polynomial time as well).

Suppose I had some Subset Sum problem asking to make T from $x_1, x_2, \ldots, x_n$. Let $S = x_1 + \ldots + x_n$ be the sum of the numbers; then I claim the result is the same as plugging into the Partition Problem using the numbers

$$x_1, x_2, \ldots, x_n, 2T - S.$$

- Since the sum of these numbers is $S + (2T - S) = 2T$, any equal partition would have each side summing to T. Since only one side has the extra $2T - S$ variable, the other side satisfies the Subset Sum.
- On the other hand, if I have a Subset $x_1, x_2, \ldots, x_i$ summing to T, then one possible partition is

$$\underbrace{x_1, x_2, \ldots, x_i}_{T} \mid \underbrace{x_{i+1}, \ldots, x_n}_{S-T}, 2T - S.$$

(By the way, if you are concerned with all the numbers being nonnegative, notice that Subset Sum with a target of T is also equivalent to Subset Sum with a target of $S - T$. So if $2T - S$ is negative, I can replace $T \leftarrow S - T$ to make it positive).

Problem 11.5.2

Make the same reduction as the previous problem, but instead consider plugging into the 3-Partition Problem with

$$x_1, x_2, \ldots, x_n, T, 2T - S.$$

- Now the total sum is $3T$, so any three equal partitions must sum to T. In a similar vein, at most two of them have one of my extra numbers added—the third is a Subset Sum of the original x_is summing to T.
- Analogously if $x_1 + x_2 + \ldots x_i = T$ then I can partition the set into three parts as follows:

$$\underbrace{x_1, x_2, \ldots, x_i}_{T} \mid T \mid \underbrace{x_{i+1}, \ldots, x_n}_{S-T}, 2T - S.$$

Problem 11.5.3

I shall show that using the any-k Partition Problem as a black box, I can solve the regular Partition Problem in polynomial time.

Suppose the Partition Problem involved $x_1, x_2, \ldots, x_n$ with a sum of S. Let $B = 2(|x_1| + |x_2| + \cdots + |x_n|) + 97$ be a huge number, and try the any-k Partition Problem with

$$x_1, x_2, \ldots, x_n, B, B.$$

The idea is that since B is so large, there must be exactly two partitions. Specifically, if there were three or more partitions, the group without a copy of B has a sum less than $\frac{B}{2}$, while a group with a copy of B has a sum more than $\frac{B}{2}$, which can't happen.

And then clearly there will be one B in each partition, which cancel each other out, so the result of the black box is equivalent to the answer of the regular Partition Problem.

Problem 11.5.4

This is trivial—given a Partition Problem on $x_1, \ldots, x_n$, the Almost Partition Problem on

$$2x_1, 2x_2, \ldots, 2x_n$$

is equivalent. (Any sum is even, so the differ-by-1 condition never happens). Therefore if a polynomial time algorithm for Almost Partition exists, then polynomial time algorithms exist for the normal Partition Problem, Subset Sum, and SAT.

Problem 11.5.5

I shall show that using the Perfect Partition Problem as a black box, I can solve the regular Partition Problem in polynomial time.

Suppose I was doing the partition problem on $x_1, \ldots, x_n$, and let $B = |x_1| + |x_2| + \cdots + |x_n| + 97$. Then, I claim the answer is equivalent to *any* of the following Perfect Partition Problems being Yes:

- $x_1, x_2, \ldots, x_n, B, B$
- $x_1, x_2, \ldots, x_n, B, B, 2B$
- $x_1, x_2, \ldots, x_n, B, B, B, 3B,$
- $\cdots$
- $x_1, x_2, \ldots, x_n, \underbrace{B, B, \ldots, B}_{n\ B\text{s}}, nB.$

I like to view the Partition Problem as having weights and trying to balance a scale—the idea here is that the Bs are heavy enough so that if the total weight of the Bs doesn't balance, it's impossible to add the other weights to correct it.

So for the ith problem, all i Bs must go on one side, and the iB weight on the other. Hence if a Perfect Partition exists for one of the problems, after removing the extra weights, I've gotten a valid solution to the Partition Problem.

For the other direction, if I had a Partition (WLOG) with $x_1 + \cdots + x_i = x_{i+1} + \cdots x_n$ and $i \leqslant \frac{n}{2}$, then in the $n - 2i + 1$th query, I have the Perfect Partition

$$x_1, x_2, \ldots, x_i, \underbrace{B, \ldots, B}_{n-2i+1\ B\text{s}} \mid x_{i+1}, \ldots, x_n, (n - 2i + 1)B$$

where both sides have equal sum and exactly $n - i + 1$ elements.

So if the Perfect Partition Problem is solvable in polynomial time, then the regular Partition Problem is too (with an extra factor of n).