Exercise 11.5 Let $x_1, \ldots, x_n \in \mathbb{N}$. For each of the following problems, either (a) design and analyze a polynomial time algorithm (the faster the better), or (b) prove that a polynomial time algorithm would imply a polynomial time algorithm for SAT.[2]

[2]You can use the solution of one subproblem to solve another, as long as there's no circular dependencies overall.

Exercise 11.5.1. The *partition problem* asks if one can partition $x_1, \ldots, x_n$ into two parts such that the sums of each part are equal.

Solution. We claim that a polynomial time solution for the partition problem would imply a polynomial time solution for SAT. To see this, we present a polynomial time reduction from subset sum, a problem known to be hard, to the partition problem.

Note (Notation)**.** Given some set $S = \{s_1, \ldots, s_n\}$, we denote the sum $\sum\limits_{s \in S} s$ with ΣS.

Consider an arbitrary instance of subset sum. That is, suppose we have a set of positive integers

$$A := \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{Z}_+$$

and a positive integer target value $T \in \mathbb{Z}_+$. Now, let $x := 2T - \Sigma A$, and define a new set $\overline{A} := A \cup \{x\}$. In the case that $T < \Sigma A/2$, notice that the value of x will end up being negative. However, if there exists some set $B = \{\beta_1, \ldots, \beta_k\} \subseteq A$ such that $\Sigma B = T$ then $A \setminus B$ sums to $\Sigma A - T \leq \Sigma A/2$, so we can simply rephrase the problem to use $\Sigma A - T$ as the target value.

Then, notice that

$$\begin{aligned} \Sigma\overline{A} &= \Sigma A + 2T - \Sigma A \\ &= 2T. \end{aligned}$$

Correctness. (SS $\implies$ PP) Suppose there exists some $B \subseteq A$ such that $\Sigma B = T$. Consider the partition of $\overline{A}$ defined as $\overline{B} = B \cup \{x\}$. Then,

$$\Sigma\overline{B} = T + 2T - \Sigma A.$$

The remaining partition is then $C := \overline{A} \setminus \overline{B}$, and

$$\begin{aligned} \Sigma C &= 2T - T + 2T - \Sigma A \\ &= T + 2T - \Sigma A, \end{aligned}$$

and we can see that these two sums are equal. Hence, the partition problem is solved.

(SS $\impliedby$ PP) Suppose the set $\overline{A}$ has a valid partition such that each of the two subsets sum to T. Recall that $\overline{A}$ is defined as the union of A and the singleton set $\{x\}$. By the pigeonhole principle, we know that one of these subsets of $\overline{A}$ is a subset of A, whence the subset sum problem is solved. ∎

Since each step in the reduction process takes only $O(1)$ or $O(n)$ time, the entire reduction can be done in polynomial time relative to the size of A. Thus, a polynomial time solution for the partition problem implies a polynomial time solution for SAT. □

Exercise 11.5.2. The *3-partition problem* asks if one can partition $x_1, \ldots, x_n$ into 3 parts such that the sums of each part are all equal.

Solution. □

Exercise 11.5.3. The *any-k-partition problem* asks if one can partition $x_1, \ldots, x_n$ into k parts, for any integer k ≥ 2, such that the sums of each part are all equal.

Solution. $\square$

Exercise 11.5.4. The *almost-partition problem* asks if one can partition $x_1, \ldots, x_n$ into two parts such that the two sums of each part differ by at most 1.

Solution. □

Exercise 11.5.5. [3]Let n be even. The *perfect partition problem* asks if one can partition $x_1, \dots, x_n$ into two parts such that

(a) Each part has the same sum.

(b) Each part contains the same number of x_i's.

[3]IMO, this one is the trickiest.

Solution. □