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. □

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 $\geq$ 2, such that the sums of each part are all equal.

Solution. □

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 $x_1, \ldots, x_n$ 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. □