Exercise 1:
Consider the following variant of the Knapsack problem:
Given \( n, c_1, \ldots, c_n, w_1, \ldots, w_n \in \mathbb{N} \) and a nonincreasing function \( b(i) : \{1, \ldots, n\} \to \mathbb{N} \),
maximize \( \sum_{i=1}^{n} c_i x_i \) subject to \( \sum_{i=1}^{n} w_i x_i \leq b(\sum_{i=1}^{n} x_i), x_i \in \{0, 1\}, i = 1, \ldots, n. \)
Show how this variant can be polynomially reduced to the standard Knapsack problem.

(4 points)

Exercise 2:
Suppose that in an instance \( a_1, \ldots, a_n \) of the Bin-Packing problem we have \( a_i > \frac{1}{3} \) for each \( i \). Reduce the problem to the Cardinality Matching problem. Then show how to solve the problem in \( O(n \log n) \) time.

(3 points)

Definition:
An absolute approximation algorithm for an optimization problem is a polynomial-time algorithm \( A \) for which there exists a constant \( k \) such that \( |A(I) - OPT(I)| \leq k \) holds for every instance \( I \).

Exercise 3:
Prove that there does not exist an absolute approximation algorithm for the Knapsack Problem (unless \( P = NP \)).

(3 points)

Exercise 4:
Find an NP-hard optimization variant of Partition for which an absolute approximation algorithm \( A \) with \( k = 1 \) exists, and prove that it exists.

(4 points)

Please return the exercises until Tuesday, May 15th, at 2:15 pm.