Problem 1

A server has $n$ customers waiting to be served. The service time required by each customer is known in advance: it is $t_i$ minutes for customer $i$. So if, for example, the customers are served in order of increasing $i$, then the $i$-th customer has to wait $\sum_{j=1}^{i-1} t_j$ minutes. We wish to minimize the total waiting time

$$T = \sum_{i=1}^{n} \text{(time spent waiting by customer i)}$$

Give an efficient algorithm for computing the optimal order in which to process the customers. Prove that your algorithm is correct and analyze its running time. (Hint: Does the greedy algorithm work in this case?)

Problem 2

Consider the following variation on the change-making problem: you are given denominations $x_1, x_2, \ldots, x_n$, and you want to make change for a value $v$, but you are allowed to use each denomination at most once. For instance, if the denominations are 1, 5, 10, 20, then you can make change for 16 = 1 + 15 and for 31 = 1 + 10 + 20 but not for 40 (because you cannot use 20 twice).

Input: Positive integers $x_1, x_2, \ldots, x_n$; another integer $v$.
Output: Can you make change for $v$, using each denomination $x_i$ at most once?

Show how to solve this problem in time $O(nv)$. 