Problem 1. Fractional Knapsack (greedy algorithm). There are n items $I_1, \cdots, I_n$. Item I_i has weight w_i and worth v_i. All these items can be broken into smaller pieces. We have a knapsack with capacity S and want to pack this knapsack with the maximum value. So we may decide to carry only a fraction x_i of item I_i where $0 \leqslant x_i \leqslant 1$. Example: if S = 10, w1 = 5, v1 = 2, w2 = 6, v1 = 1, w3 = 4, v1 = 3, then the optimal value we can get occurs when we take 4 units of the item 3 (worth 3), 5 units of item 1 (worth 2), and 1 unit of item 2 (worth 1/6) for a total value of 31/6.
Solution. Sort according to decreasing unit value: $\frac{v_1}{w_1} \geqslant \frac{v_2}{w_2} \geqslant \cdots \geqslant \frac{v_n}{w_n}$
Optimality: The optimality is trivial. There are two factors we should consider.

- There is an optimal solution that begins with the item that has the maximum unit value. To prove this, let the item I' has the maximum ratio $\frac{v'}{w'}$. Thus for all items I we have $\frac{v'}{w'} \geqslant \frac{v}{w}$. Thus, $v' \geqslant \frac{v \times w'}{w}$. Now assume there is an optimal solution that does not use the full w' weight of I'. Then if we replace any amount of any other item I in the solution with I', the total value will not decrease.

- Let O be the optimal solution for the original problem P. Moreover, let I be the first greedy choice we make in O. Then $O - I$ is the optimal solution for the subproblem P' after the first greedy choice has been made.

Problem 2. Knapsack with repetition (dynamic programming): page 167, the DPV textbook (in Section 6.4). Note the difference with the classic knapsack problem (knapsack without repetition).

Problem 3. The traveling salesman problem (dynamic programming): pages 173-175, the DPV textbook (in Section 6.6).