Up to **four** *people can work on an exercise together. But each of you should be able to explain the solutions to the TA (Bremser). Write your names* **and** *the name of your group (time, TA) on the sheets. Staple them together*

Assignment 11

Deadline: February 9, 2004

Solve at least two of the following three exercises.

Exercise 1

In this exercise we discuss how to convert a minimization problem into a decision problem. Assume that the cost function has nonnegative integer values and that we know an upper bound $\hat{f} \geq f^*$ for the value of an optimal solution. Further assume that we have a program $\mathcal{P} : \mathbb{N} \to \mathcal{L} \cup \{\perp\}$ so that $\mathcal{P}(b)$ outputs a feasible solution $\mathbf{x} \in \mathcal{L}$ with $f(\mathbf{x}) \leq b$ if such a solution exists and signals failure $\mathcal{P}(b) = \perp$ otherwise.

(a) Explain how to find an optimal solution using $O(\log \hat{f})$ calls of $\mathcal{P}$.

(b) Refine the procedure from part (a) so that it works using $O(\log f^*)$ executions of $\mathcal{P}$, even if we do not know an upper bound of f^*.

Exercise 2

Suppose we wish to invest 14,000 euros for one year. We have identified four investment opportunities. Investment 1 requires an investment of 5,000 euros and gives a profit of 800 euros; investment 2 requires 7,000 euros and gives a profit of 1,100 euros; investment 3 requires 4,000 euros and gives a profit of 600 euros; and investment 4 requires 3,000 euros and gives a profit of 400 euros. Into which investments should we place our money so as to maximize our total profit? Model the problem as a knapsack problem. (Hint: For convenience, drop extra zeros.)

Exercise 3

Consider the following algorithm for the traveling salesman problem for undirected graphs with triangle inequality (see the lecture slides for a definition):

- Let T denote the edges of a minimum spanning tree of G.
- Now consider the directed graph $G' = (V, T')$ where for each edge $\{u, v\} \in T$, T' contains the two directed edges (u, v) and (v, u). Note that each node in (V, T') has the same in-degree as out-degree.
- Find an Euler tour C' in G'.
- Convert C' into a simple cycle visiting all nodes by introducing *shortcuts*, i.e., repeatedly apply the following shortcutting step: If there are subsequent edges (u, v) and (v, w) in C such that v is also visited before, replace these two edges by the edge (u, w).

Now we ask you to show that this algorithm yields a two-approximation of the optimal TSP tour. (Hint: you might need to show that the weight of T is no more than the weight of an optimal traveling salesman tour.) Do not forget to argue the above algorithm works in polynomial time.