TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Software Systems

OHJ-2556 **Artificial Intelligence** Exercises 3
prof. Tapio Elomaa Spring 2010

1. Give the name of the algorithm that results from each of the following special cases:

   (a) Local beam search with $k = 1$.
   (b) Local beam search with $k = \infty$.
   (c) Simulated annealing with $T = 0$ at all times.
   (d) Genetic algorithm with population size $N = 1$.

2. In the *Travelling Salesperson Problem* (TSP) one is given a fully connected, weighted, undirected graph and is required to find the *Hamiltonian cycle* — a cycle visiting all of the nodes in the graph exactly once — that has the least total weight.

   Outline how hill-climbing search could be used to solve TSP. How good results would you expect hill-climbing to attain? Can other local search algorithms be used to solve TSP?

3. Let us consider the game of tic-tac-toe (noughts and crosses) on a 3×3 board. We define X_n as the number of rows, columns, or diagonals with exactly n X's and no O's. Similarly, O_n is the number of rows, columns, or diagonals with just n O's.

   The utility function assigns $+1$ to any position with $X_3 = 1$ and -1 to any position with $O_3 = 1$. All other terminal positions have utility 0. For any nonterminal position s we use a linear evaluation function $\mathit{eval}(s) = 3X_2(s) + X_1 - (3O_2(s) + O_1(s))$.

   (a) Show the whole game tree starting from the empty board down to depth 2 (i.e., one X and one O on the board), taking symmetry into account.
   (b) Mark on your tree the evaluation of all positions at depth 2.

4. Continuing with tic-tac-toe:

   (a) Using the minimax algorithm, mark on your tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.

   (b) Circle the nodes at depth 2 that would *not* be evaluated if alpha-beta pruning were applied, assuming that the nodes are generated in the optimal order for alpha-beta pruning.

5. Prove the following assertion: for every game tree, the utility obtained by MAX using minimax decisions against a suboptimal MIN will never be lower than the utility obtained playing against an optimal MIN.