# CS360 – Homework #6

## Search-Based Planning

**1)** Consider the formulation of the Tower of Hanoi puzzle as a planning problem (solution of Homework 4, problem 2). Using the delete relaxation, calculate the length of the action sequence for achieving each predicate that appears in the goal state from the start state, then calculate the heuristic value of the start state as the maximum of these values.

## Local Search

**2)** In the $N$-Queens problem, we want to place $N$ queens on an $N \times N$ board with no two queens on the same row, column, or diagonal. Come up with a value function and use hill climbing to try to solve the problem by minimizing this value function, starting with the configuration given below. Generate the successors of a state by moving a single queen vertically.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | | | |
| 2 | Q1 | | Q3 | |
| 3 | | Q2 | | Q4 |
| 4 | | | | |

**3)** How would you approach the Traveling Salesman Problem if we wanted to find a good (not necessarily the best) solution to it using hill climbing? Explain also how A* search can be used for this purpose.

**4)** What are the advantages/disadvantages of local search methods (such as hill climbing and simulated annealing) compared to A*? For which kind of optimization problems should local search be preferred?