1. Do question 4.1 in RN.

2. Do question 4.2 in RN. (Note that "objective function" means "evaluation function")

3. (Question 9.2 in Nilsson) This exercise assumes you have already completed question 2 in homework 1. Refer to your solution to that exercise. Propose an admissible $h$ function for this problem that is better than $h \equiv 0$.

4. (Question 9.3 in Nilsson) Algorithm $A^*$ does not terminate until a goal node is selected for expansion. However, a path to a goal node might be reached long before that node is selected for expansion. Why not terminate as soon as a goal node has been found? Illustrate your answer with an example.

5. For the sliding block puzzle (see hw1), specify a heuristic function $h$, and show the search tree produced by algorithm $A^*$ using this function. Show the first 10 nodes expanded.

6. Prove the following properties on algorithm $A^*$.

   (a) A heuristic function is monotone if for every node $n$ and its child node $n'$
   
   $$ h(n) \leq h(n') + c(n, n') $$

   Prove that if $h_1$ and $h_2$ are both monotone, so also is $h = \max(h_1, h_2)$.

   (b) Prove that if $h$ is monotone then it is also admissible.

   (c) Prove that if the heuristic function is monotone then $A^*$ will never reopen any nodes.

   (d) Prove or give a counter example: if for every node $n$, $h_1(n) \geq h_2(n)$, then $A^*$ with $h_1$ always expands less nodes than $A^*$ with $h_2$.  

   1
(e) Let $h$ be an admissible function and let $f(n) = w \cdot g(n) + (1 - w)h(n)$, $0 \leq w \leq 1$. Will $A*$ with $f$ find an optimal solution when $w = 1/4$? $w = 1/2$? $w = 3/4$? Can you provide a general rule? (note, that $f$ here denotes an arbitrary evaluation function, not necessarily an exact one).

7. (Extra Credit: Question 9.4 in Nilson) The monotone condition on the heuristic function requires that $h(n_i) \leq h(n_j) + c(n_i, n_j)$ for all node-successor pairs $(n_i, n_j)$, where $c(n_i, n_j)$ is the cost on the arc from $n_i$ to $n_j$. It has been suggested that when the monotone condition is not satisfied, this fact can be discovered and $h$ adjusted during search so that the condition is satisfied. The idea is that whenever a node $n_i$ is expanded, with successor node $n_j$, we can increment $h(n_j)$ by whatever amount is needed to satisfy the monotone condition. Construct an example to show that even with this scheme, when a node is expanded we have not necessarily found a least costly path to it.

8. Question 4.9 in RN.