This assignment is to be turned in individually. You may talk to others and discuss ideas, but you should ultimately turn in your own work. You should turn it in electronically. You may do it on paper and scan it in, or use some sort of software to produce it.

1. Design a search tree where iterative deepening search performs dramatically worse than depth-first search. (For example, iterative deepening might run at $O(n^2)$, whereas depth-first on that same tree might run at $O(n)$.)

2. Sometimes there is no good evaluation function for a problem, but there is a good comparison method: a way to tell if one node is better than another, without assigning numerical values to either. Show that this is enough to do a best-first search. Describe in words and pseudo code how you would implement this method, what data structure(s) you would need, and show what the differences in time and memory would be (if any) compared to a standard greedy search where you know the particular value associated with each node.

3. Consider a best-first tree-search problem in which the objective function is $f(n) = (2 - w)g(n) + wh(n)$. For what values of w is this algorithm guaranteed to be optimal, assuming that h is admissible? Prove your results. (You may assume that $0 \leq w \leq 2$). What kind of search does this perform when $w = 0$? When $w = 1$? When $w = 2$?

4. Prove that if h never overestimates by more than c, A* using h returns a solution whose cost exceeds that of the optimal solution by no more than c.