Problem 5

A heuristic h(n) is consistent if it satisfies h(n) <= c(n, a, n’) + h(n’) where c(n, a, n’) is the step cost for going from n to n’ using action a. Let k to be the number of nodes on the shortest path to n’ from n, and h\*(n) be the true cost from n to n’.

Base Case: when k = 1, h(n) <= c(n, a, n’) + h(n’) = c(n, a, n’) = h\*(n)

Inductive Hypothesis: Assume h(n’) is admissible when k = i

Inductive Step: h(n) <= c(n, a, n’) + h(n’) = c(n, a, n’) + h\*(n’) = h\*(n) since h(n) <= h\*(n). Since h(n) <= h\*(n), h(n) is admissible when k = i+1

It is proved by induction that when a heuristic is consistent, it must be admissible.

However, an admissible heuristic is not always consistent.

For a heuristic to be admissible it must satisfy h(n) <= h\*(n). If there is a triangle with three nodes s (start state), x, g (goal state). Let s to x takes 2 cost, and x to g takes 4 costs. then h(s) <= 6, h(x) <= 4, h(g) <= 0. If this case is consistent, h(s) <= 2 + h(x) and h(x) <= 4 + h(g). A heuristic that satisfies h(s) > 2 + h(x) and h(x) <= 4 is inconsistent. If h(x) = 2, it is admissible but not consistent.

Problem 6

In a CSP search, it is a good heuristic to choose the variable that is most constrained because it has the highest chance of causing a failure and it is more efficient to fail as early as possible. If we choose the least constrained value, it has the lowest chance of causing a failure and there will be more assignments taking place.