Proof of Consistency for cornersHeuristic

- Definition of states and costs:

Let n be the current state, n = (pos_n, visited_n).

Let n' be a successor state, n' = (pos_n', visited_n'), reached by a single action.

The cost of the action c(n, n') is always 1 in this problem.

The set of unvisited corners at n' is U(n'), which is either the same as U(n) or is U(n) minus the corner at pos_n'. Therefore, U(n') is a subset of U(n).

- Application of Manhattan Distance's Triangle Inequality:

The Manhattan distance itself satisfies the triangle inequality. For any three points a, b, and z:

manhattanDistance(a, z) <= manhattanDistance(a, b) + manhattanDistance(b, z)

Let a = pos_n, b = pos_n', and z = c, where c is any unvisited corner in U(n'). Then:

manhattanDistance(pos_n, c) <= manhattanDistance(pos_n, pos_n') + manhattanDistance(pos_n', c)

Since pos_n' is a direct neighbor of pos_n, the Manhattan distance between them is 1.

manhattanDistance(pos_n, pos_n') = 1

So, for any corner c in U(n'):

manhattanDistance(pos_n, c) <= 1 + manhattanDistance(pos_n', c)

Relate Heuristics of n and n':

The heuristic h(n') is the maximum Manhattan distance from pos_n' to any corner in U(n'):

h(n') = max_{c in U(n')} manhattanDistance(pos_n', c)

From our inequality in step 2, we know that for any c in U(n'):

manhattanDistance(pos_n, c) <= 1 + h(n')

This is because h(n') is the maximum of the right-hand term manhattanDistance(pos_n', c).

Since this holds for every corner c in U(n'), it must also hold for the corner that is furthest from pos_n:

max_{c in U(n')} manhattanDistance(pos_n, c) <= 1 + h(n')

Final Step - Consider h(n):

The heuristic h(n) is max_{c in U(n)} manhattanDistance(pos_n, c). Since U(n') is a subset of U(n), the maximum distance over the larger set U(n) must be greater than or equal to the maximum distance over the smaller set U(n').

h(n) = max_{c in U(n)} manhattanDistance(pos_n, c) >= max_{c in U(n')} manhattanDistance(pos_n, c)

Combining this with the result from step 3:

h(n) >= max_{c in U(n')} manhattanDistance(pos_n, c)

And we know:

max_{c in U(n')} manhattanDistance(pos_n, c) <= 1 + h(n')

This does not directly prove h(n) <= 1 + h(n'). Let's re-approach.

Let c_max be the corner in U(n) that is furthest from pos_n. So, h(n) = manhattanDistance(pos_n, c_max).

h(n) = manhattanDistance(pos_n, c_max) <= manhattanDistance(pos_n, pos_n') + manhattanDistance(pos_n', c_max) (by triangle inequality)

h(n) <= 1 + manhattanDistance(pos_n', c_max)

Now we compare manhattanDistance(pos_n', c_max) with h(n').

Case 1: c_max is also in U(n'). In this case, manhattanDistance(pos_n', c_max) is one of the values considered when calculating h(n'). By definition, it must be less than or equal to the maximum value, so manhattanDistance(pos_n', c_max) <= h(n').

Therefore, h(n) <= 1 + h(n').

Case 2: c_max is not in U(n'). This only happens if pos_n' = c_max. In this case, h(n') is the max distance from c_max to the other unvisited corners. But h(n) is the distance from pos_n to c_max, which is 1. So h(n) = 1. Since all heuristic values are non-negative (h(n') >= 0), the inequality 1 <= 1 + h(n') holds.

In all cases, the condition h(n) <= 1 + h(n') is satisfied. Thus, the heuristic is consistent.