Assignment 1

Garrett Kinman – 260763260

Question 1: Six Puzzle

Part a)

With BFS and UCS, total solution cost of 3, solution found at depth 3. The results, as shown in the IDE workspace, are shown below. In the interests of brevity, the solution path for DFS is not shown, as the path it found had a solution depth and cost of 163. Iterative deepening also found a solution cost of 3 and depth 3. This demonstrates how DFS does not guarantee optimality. Because there are $6! = 720$ possible states, the DFS algorithm could keep on moving, not getting to a solution for a while, but not seeing previous states either.

Figure 1: Solution path for breadth-first search; top-level node represents goal, and down represents the chain of parents.

Figure 2: Solution path for uniform-cost search; top-level node represents goal, and down represents the chain of parents.

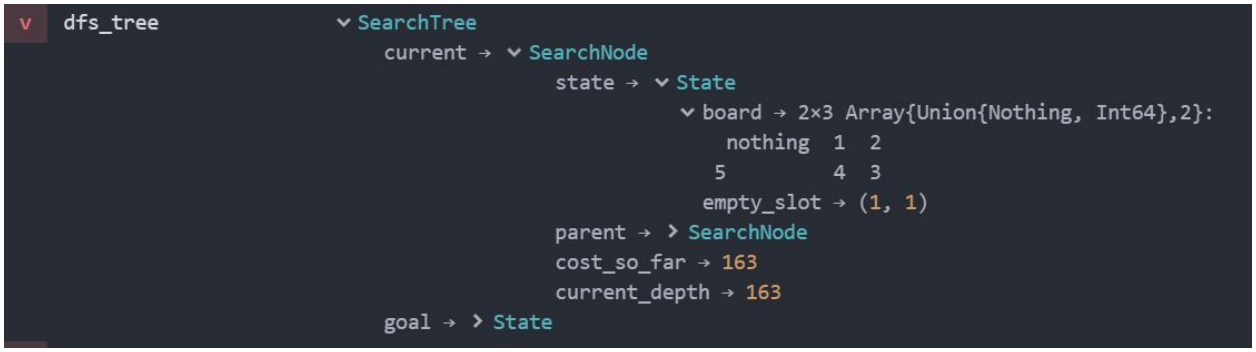

Figure 3: Terminus of solution path for depth-first search (full path omitted for brevity).

Figure 4: Solution path for iterative deepening search; top-level node represents goal, and down represents the chain of parents.

Part b)

The key constraint on an admissible heuristic is that it is *optimistic*, i.e., $h(n) \leq h^*(n), \forall n$, where $h^*(n)$ is the shortest path from n to any goal state. In the case of Manhattan distance heuristic, we already know it is admissible in the case of unit cost, meaning the inequality holds. If we instead associate the cost of a given transition to the number of the tile—which is always ≥1—instead of unit cost, the shortest path from n to any goal state is, in the best case, the same as for unit cost. Else, it is higher. Because this stipulation only modifies the RHS of the above equation, Manhattan distance must still be an admissible heuristic.

Part c)

An admissible heuristic that dominates the above for this scenario is a like modification to Manhattan distance. Instead of unit cost per position away from goal per tile, have that cost also be the same as the tile number. Essentially, if tile 1 is two spots away from goal, it contributes $1 \times 2 = 2$ cost, and if tile 4 is 3 spots away from goal, it contributes $4 \times 3 = 12$ cost. This heuristic dominates Manhattan distance in this case, as, for the same state, each tile will contribute at least the same cost towards the heuristic value at that state. And all but tile 1 contribute more cost per tile. Meaning this heuristic's cost will always be greater than or equal

to Manhattan distance's cost. Also, it is admissible for the same reason Manhattan distance is admissible for unit cost.

Question 2: Search Algorithms

Part a)

Suppose a long single chain of states, each with only one successor state, n states in total. DFS would visit each one once for a time complexity of O(n). Iterative deepening search, however, would visit $1 + 2 + 3 + ... + n - 1 + n = \sum_{k=1}^{n} k = \frac{1}{2}n(n+1)$ nodes, making it $O(n^2)$.

Part b)

TODO

Part c)

TODO

Part d)

TODO

Part e)

TODO

Question 3: Travelling Salesman Problem

Part a)

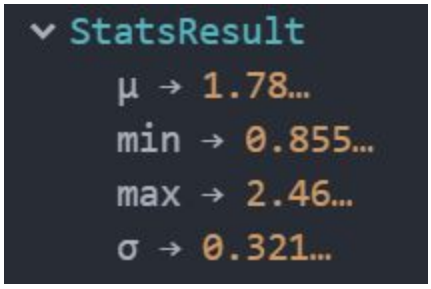

Figure 5: Mean, min, max, and standard deviation for optimal seven-city TSPs tours.

Part b)

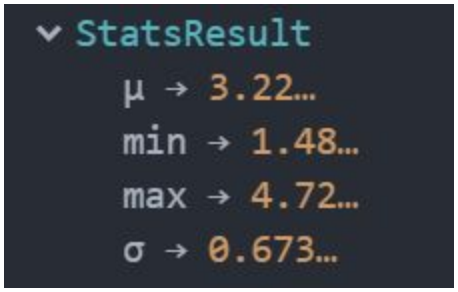

Figure 6: Mean, min, max, and standard deviation for random seven-city TSPs tours.

Additionally, none of the random tours were optimal.

Part c)

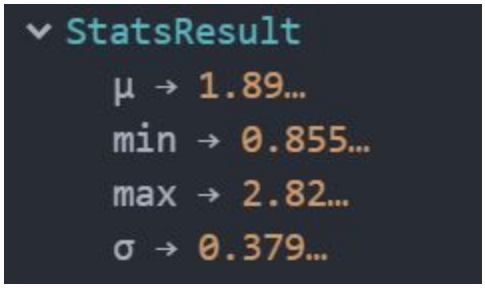

Figure 7: Mean, min, max, and standard deviation for hill-climbed seven-city TSPs tours.

Additionally, 38 of 100 tours were optimal.

Part d)

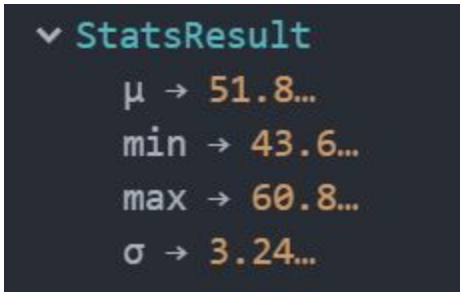

Figure 8: Mean, min, max, and standard deviation for random 100-city TSPs tours.

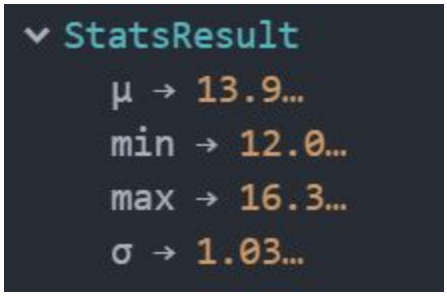

Figure 9: Mean, min, max, and standard deviation for hill-climbed 100-city TSPs tours.

Packages Used

[1] Statistics.jl (for *mean*, *std* functions)
[2] Combinatorics.jl (for combinations and permutations)
[3] Pipe.jl (for the *@pipe* macro, a general utility)