7.1 Table of Contents

Profiles

Iterative Deepening Search:

```
────────────────────────────────────────────────────────────────────────────────────────────────────
────────────────────────────────
                                        Time                   Allocations
                                ─────────────────────────────────────
──────────────────────────────────────
        Tot / % measured:           5.07ms / 69.6%           1.07MiB / 97.5%

 Section                ncalls     time   %tot     avg     alloc   %tot      avg
────────────────────────────────────────────────────────────────────────────────────────────────────
────────────────────────────────
 depth limited search        5   3.53ms   100%   706μs   1.04MiB  100%    214KiB
   expand                  102   1.55ms  44.0%  15.2μs    640KiB  59.9%  6.27KiB
   possibleactions         102   1.04ms  29.4%  10.2μs    379KiB  35.4%  3.71KiB
   create node             284    224μs  6.36%   790ns   31.1KiB  2.91%     112B
   push                    568    148μs  4.20%   261ns   5.69KiB  0.53%    10.3B
   pop                     283   51.0μs  1.45%   180ns     0.00B  0.00%    0.00B
```

Breadth-First Search:

```
────────────────────────────────────────────────────────────────────────────────────────────────────
───────────────────────
                                   Time                   Allocations
                           ─────────────────────────────────────
──────────────────────────────────────
     Tot / % measured:          1.06ms / 39.2%           642KiB / 95.1%

 Section           ncalls     time   %tot     avg     alloc   %tot      avg
```

```
──────────────────────────────────────────────────────────────────────────────────────────
 expand               59    236μs  56.8%  3.99μs    375KiB  61.5%  6.36KiB
 possibleactions      59    136μs  32.9%  2.31μs    219KiB  35.9%  3.72KiB
 create node         109   25.5μs  6.13%   234ns   11.9KiB  1.95%     112B
 push                218   13.8μs  3.33%  63.4ns   4.25KiB  0.70%    20.0B
 pop                  59   3.71μs  0.89%  62.9ns     0.00B  0.00%    0.00B
──────────────────────────────────────────────────────────────────────────────────────────
```

A* Search with Manhattan Distance

```
──────────────────────────────────────────────────────────────────────────────────────────
                                Time                    Allocations
                          ──────────────────────   ───────────────────────
     Tot / % measured:          759μs / 18.5%            116KiB / 69.6%

 Section            ncalls     time   %tot     avg     alloc   %tot      avg
──────────────────────────────────────────────────────────────────────────────────────────
 expand                7   56.9μs  40.6%  8.13μs   45.3KiB  56.2%  6.47KiB
 heuristic score      19   37.9μs  27.0%  1.99μs   7.13KiB  8.84%     384B
 possibleactions       7   28.6μs  20.4%  4.08μs   26.0KiB  32.3%  3.72KiB
 create node          19   8.24μs  5.88%   434ns   2.08KiB  2.58%     112B
 dequeue               7   7.80μs  5.56%  1.11μs     0.00B  0.00%    0.00B
 push                  7    695ns  0.50%  99.3ns      128B  0.16%    18.3B
──────────────────────────────────────────────────────────────────────────────────────────
```

A* Search with Manhattan Distance and Linear Conflicts

```
──────────────────────────────────────────────────────────────────────────────────────────
                                Time                    Allocations
                          ──────────────────────   ───────────────────────
     Tot / % measured:         30.3ms / 98.2%            469KiB / 92.5%

 Section            ncalls     time   %tot     avg     alloc   %tot      avg
──────────────────────────────────────────────────────────────────────────────────────────
 heuristic score      19   29.7ms   100%  1.56ms    360KiB  83.0%  18.9KiB
 possibleactions       7   37.4μs  0.13%  5.35μs   26.0KiB  6.00%  3.72KiB
 expand                7   36.1μs  0.12%  5.16μs   45.3KiB  10.4%  6.47KiB
 create node           19   9.86μs  0.03%   519ns   2.08KiB  0.48%     112B
 dequeue                7   8.52μs  0.03%  1.22μs     0.00B  0.00%    0.00B
 push                   7    773ns  0.00%   110ns      128B  0.03%    18.3B

────────────────────────────────────────────────────────────────────────────────────────────────────
───────────────────────
```

Solutions

Iterative Deepening Search:

```
Step 6:
        [0, 1, 2]
        [3, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(1, 1), "DOWN")
Step 5:
        [3, 1, 2]
        [0, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 4:
        [3, 1, 2]
        [4, 0, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 2), "DOWN")
Step 3:
        [3, 1, 2]
        [4, 7, 5]
        [6, 0, 8]
        Action: (CartesianIndex(3, 2), "LEFT")
Step 2:
        [3, 1, 2]
        [4, 7, 5]
        [0, 6, 8]
        Action: (CartesianIndex(3, 1), "UP")
Step 1:
        [3, 1, 2]
        [0, 7, 5]
        [4, 6, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 0:
        [3, 1, 2]
        [7, 0, 5]
        [4, 6, 8]
        Action: nothing
```

Breadth-First Search:

```
Step 6:
        [0, 1, 2]
        [3, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(1, 1), "DOWN")
Step 5:
        [3, 1, 2]
        [0, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 4:
        [3, 1, 2]
        [4, 0, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 2), "DOWN")
```

```
Step 3:
        [3, 1, 2]
        [4, 7, 5]
        [6, 0, 8]
        Action: (CartesianIndex(3, 2), "LEFT")
Step 2:
        [3, 1, 2]
        [4, 7, 5]
        [0, 6, 8]
        Action: (CartesianIndex(3, 1), "UP")
Step 1:
        [3, 1, 2]
        [0, 7, 5]
        [4, 6, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 0:
        [3, 1, 2]
        [7, 0, 5]
        [4, 6, 8]
        Action: nothing
```

A* Search with Manhattan Distance

```
Step 6:
        [0, 1, 2]
        [3, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(1, 1), "DOWN")
Step 5:
        [3, 1, 2]
        [0, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 4:
        [3, 1, 2]
        [4, 0, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 2), "DOWN")
Step 3:
        [3, 1, 2]
        [4, 7, 5]
        [6, 0, 8]
        Action: (CartesianIndex(3, 2), "LEFT")
Step 2:
        [3, 1, 2]
        [4, 7, 5]
        [0, 6, 8]
        Action: (CartesianIndex(3, 1), "UP")
Step 1:
        [3, 1, 2]
        [0, 7, 5]
        [4, 6, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 0:
        [3, 1, 2]
        [7, 0, 5]
        [4, 6, 8]
        Action: nothing
```

A* Search with Manhattan Distance and Linear Conflicts

```
Step 6:
        [0, 1, 2]
        [3, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(1, 1), "DOWN")
Step 5:
        [3, 1, 2]
        [0, 4, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 4:
        [3, 1, 2]
        [4, 0, 5]
        [6, 7, 8]
        Action: (CartesianIndex(2, 2), "DOWN")
Step 3:
        [3, 1, 2]
        [4, 7, 5]
        [6, 0, 8]
        Action: (CartesianIndex(3, 2), "LEFT")
Step 2:
        [3, 1, 2]
        [4, 7, 5]
        [0, 6, 8]
        Action: (CartesianIndex(3, 1), "UP")
Step 1:
        [3, 1, 2]
        [0, 7, 5]
        [4, 6, 8]
        Action: (CartesianIndex(2, 1), "RIGHT")
Step 0:
        [3, 1, 2]
        [7, 0, 5]
        [4, 6, 8]
        Action: nothing
```