

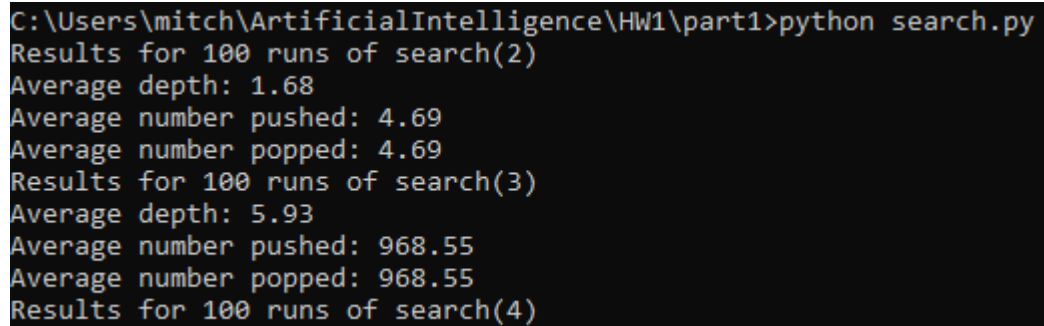
# AI HW 1

Michael Edelman

Sep 2022

## 1 HW Part 1 Results

### 1.1 Results Image



```
C:\Users\mitch\ArtificialIntelligence\HW1\part1>python search.py
Results for 100 runs of search(2)
Average depth: 1.68
Average number pushed: 4.69
Average number popped: 4.69
Results for 100 runs of search(3)
Average depth: 5.93
Average number pushed: 968.55
Average number popped: 968.55
Results for 100 runs of search(4)
```

### 1.2 Observations

for search(2) and search(3) both were executable and completed their 100 runs in reasonable amounts of time.

The Average Depth, pushed, and popped increased with the increase in N (2 to 3). This makes sense as the greater N is the greater the number of possibilities that need to be searched to find the goal state. As far as search(4) goes; many search(4) executions will not complete in a bearable amount of time. At first I thought this was my fault and something I messed up in the code, but after thinking about it: the number of possibilities and therefore the upper bound of the problem is way more than search(3) - like waaay more  $16!/9! = 57657600$  times slower. No wonder it doesn't finish - in fact if we estimate that search(3) takes one second to complete search(4) can take almost 2 years to complete!!!!

## 2 HW Part 2 Results

### 2.1 Results Images

#### 2.1.1 Step 1

```
Average Path Cost: 16.61
Average number pushed: 258746.15
Average number popped: 123661.71
```

#### 2.1.2 Step 2

```
Results for 100 runs of search(4) using the h2 heuristic (Manhattan Distance)
Average Path Cost: 16.2
Average number pushed: 1478.43
Average number popped: 705.14
C:\Users\mitch\ArtificialIntelligence\HW1\part2>
```

#### 2.1.3 Step 3

```
Results for 100 runs of search(4) using the h1 heuristic (bricks out of place) with A* weight
Average Path Cost: 16.97
Average number pushed: 23698.55
Average number popped: 11074.34
C:\Users\mitch\ArtificialIntelligence\HW1\part2>python search.py
Results for 100 runs of search(4)
Results for 100 runs of search(4) using the h2 heuristic (Manhattan Distance) with A* weight
Average Path Cost: 17.47
Average number pushed: 1989.27
Average number popped: 935.53
C:\Users\mitch\ArtificialIntelligence\HW1\part2>
```

### 2.2 Observations

The first set of results in step 1 using the number of bricks out of place heuristic took a long time to produce and go through all 100 steps. However, unlike Iterative Depth Search in part 1, using this heuristic we were able to produce and complete 100 iterations of search(4) (in around an hour). The second set of results in step 2 using the Manhattan Distance heuristic was much faster and took maybe a minute. In step 3, noticeably the h1 heuristic went through almost 10 times less states than without the A\* weight. the h2 heuristic however did not seem to change significantly with the A\* weight.