Results – Planning Search

# Daniel Meechan

## Udacity Artificial Intelligence Nanodegree

## Term 1, Project 3 (November 2017)

A\* Search with levelsum error:

Solving Air Cargo Problem 1 using astar\_search with h\_pg\_levelsum...

Traceback (most recent call last):

File "run\_search.py", line 129, in <module>

main(list(sorted(set(args.problems))), list(sorted(set((args.searches)))))

File "run\_search.py", line 101, in main

run\_search(\_p, s, \_h)

File "run\_search.py", line 57, in run\_search

node = search\_function(ip, parameter)

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/aimacode/search.py", line 276, in astar\_search

return best\_first\_graph\_search(problem, lambda n: n.path\_cost + h(n))

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/aimacode/search.py", line 213, in best\_first\_graph\_search

frontier.append(node)

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/aimacode/utils.py", line 602, in append

heapq.heappush(self.\_queue, (self.priorityFn(item), item))

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/aimacode/utils.py", line 283, in memoized\_fn

val = fn(obj, \*args)

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/aimacode/search.py", line 276, in <lambda>

return best\_first\_graph\_search(problem, lambda n: n.path\_cost + h(n))

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/aimacode/utils.py", line 283, in memoized\_fn

val = fn(obj, \*args)

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/my\_air\_cargo\_problems.py", line 214, in h\_pg\_levelsum

pg = PlanningGraph(self, node.state)

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/my\_planning\_graph.py", line 220, in \_\_init\_\_

self.create\_graph()

File "/Users/daniel.meechanibm.com/Code/Python/AI-Nanodegree/Planning-Project/my\_planning\_graph.py", line 284, in create\_graph

self.update\_a\_mutex(self.a\_levels[level])

IndexError: list index out of range

Problem 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Search algorithm | Expansions | Goal Tests | Plan length | Time |
| Breath first search | 43 | 56 | 6 | 0.031 |
| Breath first tree search | 1458 | 1459 | 6 | 0.825 |
| Depth first graph search | 12 | 13 | 12 | 0.009 |
| Depth limited search | 101 | 271 | 50 | 0.085 |
| Uniform cost search | 55 | 57 | 6 | 0.040 |
| Recursive best first search | 4229 | 4230 | 6 | 2.870 |
| Greedy best first graph search | 7 | 9 | 6 | 0.004 |
| A\* Search | 55 | 57 | 6 | 0.050 |
| A\* Search with ignore preconditions | 55 | 57 | 6 | 0.044 |

Due to the higher search time required for problems 2 and 3, I chose to test only 5 of the search algorithms those two problems.

Problem 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Search algorithm | Expansions | Goal Tests | Plan length | Time |
| Breath first search | 3343 | 4609 | 9 | 7.141 |
| Depth first graph search | 582 | 583 | 575 | 2.887 |
| Uniform cost search | 4853 | 4855 | 9 | 11.222 |
| Greedy best first graph search | 998 | 1000 | 21 | 2.125 |
| A\* Search | 4853 | 4855 | 9 | 11.003 |

Problem 3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Search algorithm | Expansions | Goal Tests | Plan length | Time |
| Breath first search | 12358 | 15077 | 12 | 32.629 |
| Depth first graph search | 4315 | 4316 | 815 | 18.898 |
| Uniform cost search | 15601 | 15603 | 12 | 41.816 |
| Greedy best first graph search | 5297 | 5299 | 24 | 13.334 |
| A\* Search | 15601 | 15603 | 12 | 40.815 |