**Logic and Planning**

**Uninformed Search Tests**

The Following are the results of tests using uninformed search.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem 1** | **Expansions** | **Goal Tests** | **New Nodes** | **Plan Lenght** | **Time Elapsed** |
| Breath First Search | 43 | 56 | 180 | 6 | 0.0743535709916614 |
| Depth First Graph Search | 12 | 13 | 48 | 12 | 0.01335207000374794 |
| A\* Search H1 | 55 | 57 | 224 | 6 | 0.07291120200534351 |
| Uniform Cost Search | 55 | 5 | 224 | 6 | 0.10046831099316478 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem 2** | **Expansions** | **Goal Tests** | **New Nodes** | **Plan Lenght** | **Time Elapsed** |
| Breath First Search | 3343 | 4609 | 30509 | 9 | 20.04158652901242 |
| Depth First Graph Search | 476 | 477 | 4253 | 466 | 3.575219680002192 |
| A\* Search H1 | 4853 | 4855 | 44041 | 9 | 77.61358179600211 |
| Uniform Cost Search | 4853 | 4855 | 44041 | 9 | 77.85962436001864 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem 3** | **Expansions** | **Goal Tests** | **New Nodes** | **Plan Lenght** | **Time Elapsed** |
| Breath First Search | 14663 | 18098 | 129631 | 12 | 198.60475270700408 |
| Depth First Graph Search | 1511 | 1512 | 12611 | 1442 | 22.499199126003077 |
| A\* Search H1 | 18223 | 18225 | 159618 | 12 | 713.9446880209871 |
| Uniform Cost Search | 18223 | 18225 | 159618 | 12 | 698.5272629769752 |

**Analysis:**

All algorithms seem to perform well with **Problem 1**, but **Problem 2** and **Problem 3** running times are definitely higher. One of the things were mentioned in the video lectures about search was that for some route finding problems DFS could be innefective because we could end up traveling all accross the country just to find the best route from two nearby cities. We can see something similar if we check the plan length for DFS. It is larger than others and doesn’t seem to be an optimal plan. In therms of Plan lenght we can say that other algorithms tested are better.

**Informed Search Problems**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem 1** | **Expansions** | **Goal Tests** | **New Nodes** | **Plan Lenght** | **Time Elapsed** |
| A\* Search Ignore Preconditions | 41 | 43 | 170 | 6 | 0.1278501100023277 |
| A\* Planning Graph Level Sum | 8 | 10 | 35 | 6 | 1.875899639009731 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem 2** | **Expansions** | **Goal Tests** | **New Nodes** | **Plan Lenght** | **Time Elapsed** |
| A\* Search Ignore Preconditions | 1506 | 1508 | 13280 | 9 | 29.742081367992796 |
| A\* Planning Graph Level Sum | 17 | 19 | 162 | 9 | 45.7323810380185 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem 3** | **Expansions** | **Goal Tests** | **New Nodes** | **Plan Lenght** | **Time Elapsed** |
| A\* Search Ignore Preconditions | 5118 | 5120 | 45650 | 12 | 183.82016788300825 |
| A\* Planning Graph Level Sum | 18 | 20 | 169 | 14 | 78.21723176701926 |

**Analysis:**

A\* Planning Graph with Level Sum heuristic performed much better than Ignore Preconditions. Less expansions, nodes and goal tests were needed, even if the plan lenght is bigger by two on Problem 3, it is an optimal plan. According to AIMA, level sum returns the sum of the level cost where any literal of the goal first appears, which is innacurate but works better than the rest.

Theoretically a better heuristic that is not included in the list, nor was tested is the **set level heuristic**, which takes the minimum level where all the literals of the goal appear and are free of mutex.

**Optimal Plans**

**Problem 1 (BFS):**

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

**Problem 2 (A\* Level Sum):**

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

**Problem 3(A\* Level Sum):**

Load(C1, P1, SFO)

Fly(P1, SFO, ORD)

Load(C2, P2, JFK)

Fly(P2, JFK, ATL)

Load(C3, P2, ATL)

Fly(P2, ATL, SFO)

Unload(C2, P2, SFO)

Fly(P2, SFO, JFK)

Load(C4, P1, ORD)

Fly(P1, ORD, JFK)

Unload(C1, P1, JFK)

Fly(P1, JFK, SFO)

Unload(C4, P1, SFO)

Unload(C3, P2, JFK)