# **Heuristic Analysis**

## Problem 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Search Method | *Processing Time* | *Plan Length* | *Expansions* | *Goal Test* | *New Nodes* |
| 1 | Breadth-First | 0.054 | 6 | 43 | 56 | 180 |
| 2 | Breadth-First Tree | 1.528 | 6 | 1,458 | 1,459 | 5,960 |
| 3 | Depth-First Graph | 0.013 | 12 | 12 | 13 | 48 |
| 4 | Depth Limited | 0.133 | 50 | 101 | 271 | 414 |
| 5 | Uniform Cost | 0.060 | 6 | 55 | 57 | 224 |
| 6 | Recursive Best First with *h\_1* | 4.446 | 6 | 4,229 | 4,230 | 17,029 |
| 7 | Greedy Best First with *h\_1* | 0.009 | 6 | 7 | 9 | 28 |
| 8 | A\* with *h\_1* | 0.062 | 6 | 55 | 57 | 224 |
| 9 | A\* with *h\_ignore\_preconditions* | 0.047 | 6 | 41 | 43 | 170 |
| 10 | A\* with *h\_peg\_levelsum* | 1.940 | 6 | 41 | 43 | 167 |

|  |  |
| --- | --- |
| Methods | Optimal Solution |
| * **Non-heuristic**: As expected, depth-first methods are fast but very ineffective. Breadth-First and the Uniform Cost searches are capable of finding optimal solutions with good timing. * **Heuristic**: All methods deliver optimal results, with a significant speed advantage for Greedy Best First graph search. Recursive Best First search was the most inefficient of them. | The best solution is the ***Greedy Best First Graph Search with h\_1***. But since the algorithm is very goal-oriented but the h\_1 heuristics is not much of an heuristic at all, this performance is probably because is very small and simple.   * Load(C1, P1, SFO) * Load(C2, P2, JFK) * Fly(P1, SFO, JFK) * Fly(P2, JFK, SFO) * Unload(C1, P1, JFK) * Unload(C2, P2, SFO) |

## Problem 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Search Method | *Processing Time* | *Plan Length* | *Expansions* | *Goal Test* | *New Nodes* |
| 1 | Breadth-First | 17.506 | 9 | 3,343 | 4,609 | 30,509 |
| 2 | Breadth-First Tree | *too long* | *-* | *-* | *-* | *-* |
| 3 | Depth-First Graph | 3.847 | 575 | 582 | 583 | 5,211 |
| 4 | Depth Limited | *too long* | *-* | *-* | *-* | *-* |
| 5 | Uniform Cost | 18.543 | 9 | 4,852 | 4,854 | 44,030 |
| 6 | Recursive Best First with *h\_1* | *too long* | *-* | *-* | *-* | *-* |
| 7 | Greedy Best First with *h\_1* | 3.629 | 21 | 990 | 992 | 8,910 |
| 8 | A\* with *h1* | 17.903 | 9 | 4,852 | 4,854 | 44,030 |
| 9 | A\* with *h\_ignore\_preconditions* | 5.956 | 9 | 1,450 | 1,452 | 13,303 |
| 10 | A\* with *h\_peg\_levelsum* | 201.928 | 9 | 86 | 88 | 841 |

|  |  |
| --- | --- |
| Methods | Optimal Solution |
| * **Non-heuristic**: 2 of the non-heuristic models timed out. Again, Breadth-First and Uniform Cost had a good performance, comparable to Heuristic methods. * **Heuristic**: Among the Heuristic methods, only the ones based in A\* were effective. Recursive best First search timed out, and Greedy Best First has a bad plan output. | The best solution is the ***A\* with h\_ignore\_preconditions*** heuristics. The algorithm was fast with an efficient plan. The A\* with the heuristics *h\_peg\_levelsum* was efficient but took too long.     * Load(C3, P3, ATL) * Fly(P3, ATL, SFO) * Unload(C3, P3, SFO) * Load(C2, P2, JFK) * Fly(P2, JFK, SFO) * Unload(C2, P2, SFO) * Load(C1, P1, SFO) * Fly(P1, SFO, JFK) * Unload(C1, P1, JFK) |

## Problem 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Search Method | *Processing Time* | *Plan Length* | *Expansions* | *Goal Test* | *New Nodes* |
| 1 | Breadth-First | 130.050 | 12 | 14,663 | 18,098 | 129,631 |
| 2 | Breadth-First Tree | *too long* | *-* | *-* | *-* | *-* |
| 3 | Depth-First Graph | 4.464 | 596 | 627 | 628 | 5,176 |
| 4 | Depth Limited | *too long* | *-* | *-* | *-* | *-* |
| 5 | Uniform Cost | 87.572 | 12 | 18,223 | 18,225 | 159,618 |
| 6 | Recursive Best First with *h\_1* | *too long* | *-* | *-* | *-* | *-* |
| 7 | Greedy Best First with *h\_1* | 25.894 | 22 | 5,578 | 5,580 | 49,150 |
| 8 | A\* with *h1* | 91.697 | 12 | 18,223 | 18,225 | 159,618 |
| 9 | A\* with *h\_ignore\_preconditions* | 24.554 | 12 | 5,040 | 5,042 | 44.944 |
| 10 | A\* with *h\_peg\_levelsum* | 1,097.362 | 12 | 325 | 327 | 3,002 |

|  |  |
| --- | --- |
| Methods | Optimal Solution |
| * **Non-heuristic**: As before, 2 of the non-heuristic models timed out. Breadth-First and Uniform Cost found a good plan, but they took a lot more time than most Heuristics methods. * **Heuristic**: As before, the A\* methods outperformed Among the Heuristic methods, only the ones based in A\* were effective. Recursive Best First search timed out again. Greedy Best First performed relatively better than before, but it’s still bad. | The best solution is again the ***A\* with h\_ignore\_preconditions*** heuristics. Again, the A\* with the heuristics *h\_peg\_levelsum* was efficient too but took a lot more time to calculate.   * Load(C2, P2, JFK) * Fly(P2, JFK, ORD) * Load(C4, P2, ORD) * Fly(P2, ORD, SFO) * Unload(C4, P2, SFO) * Load(C1, P1, SFO) * Fly(P1, SFO, ATL) * Load(C3, P1, ATL) * Fly(P1, ATL, JFK) * Unload(C3, P1, JFK) * Unload(C2, P2, SFO) * Unload(C1, P1, JFK) |