**Artificial Intelligence**

**Project 1**

**Team 3**

**8-Puzzle (Default Board State):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm (+Heuristic)** | **Nodes Expanded** | **Nodes Generated** | **Solution Cost** | **Time Taken (sec)** |
| **DFS** | - | - | - | - |
| **BFS** | 1271285 | 3589819 | 14 | 11.88 |
| **IDA** | 1965733 | 1965720 | 14 | 3.31 |
| **Djikstra** | 1163507 | 3131259 | 14 | 21.15 |
| **Greedy (Manhattan)** | - | - | - | - |
| **Greedy (Hamming)** | - | - | - | - |
| **A\* (Manhattan)** | 906 | 2439 | 14 | 0.02 |
| **A\* (Hamming)** | 5640 | 16626 | 14 | 0.10 |

DFS: Not Possible (Infinite Search Tree)

Greedy (Manhattan): My poor laptop (4.6Ghz with 16gb ram) cant handle it </3 -> (Heursitic bad and inf search tree)

Greedy (Hamming): My poor laptop (4.6Ghz with 16gb ram) cant handle it </3

**8-Puzzle (New Board State below):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm (+Heuristic)** | **Nodes Expanded** | **Nodes Generated** | **Solution Cost** | **Time Taken (sec)** |
| **DFS** | - | - | - | - |
| **BFS** | 288645 | 819651 | 12 | 1.6388 |
| **IDA** | 446963 | 446952 | 12 | 0.7711 |
| **Djikstra** | 275108 | 737463 | 12 | 2.9916 |
| **Greedy (Manhattan)** | - | - | - | - |
| **Greedy (Hamming)** | - | - | - | - |
| **A\* (Manhattan)** | 68 | 183 | 12 | 0.0020 |
| **A\* (Hamming)** | 2101 | 5969 | 12 | 0.0329 |

DFS: Not Possible (Infinite Search Tree)

Greedy (Manhattan): My poor laptop (4.6Ghz with 16gb ram) cant handle it </3

Greedy (Hamming): My poor laptop (4.6Ghz with 16gb ram) cant handle it </3

New Board State:

[1,6,5,

4,0,2,

7,3,8]

**Pacman (Default Board)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm (+Heuristic)** | **Nodes Expanded** | **Nodes Generated** | **Solution Cost** | **Time Taken (sec)** |
| **DFS** | - | - | - | - |
| **BFS** | 16707 | 36555 | 14 | 0.1139 |
| **IDA** | 30677 | 30665 | 14 | 0.0567 |
| **Djikstra** | 16723 | 36711 | 14 | 0.1944 |
| **Greedy (Euclidean)** | - | - | - | - |
| **A\* (Euclidean)** | 221 | 479 | 14 | 0.0020 |

DFS: Not Possible (Infinite Search Tree)

Greedy (Euclidean): My poor laptop (4.6Ghz with 16gb ram) cant handle it </3

**Pacman (Custom Board Below)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm (+Heuristic)** | **Nodes Expanded** | **Nodes Generated** | **Solution Cost** | **Time Taken (sec)** |
| **DFS** | - | - | - | - |
| **BFS** | 145830 | 361141 | 16 | 1.4562 |
| **IDA** | 249699 | 249685 | 16 | 0.4206 |
| **Djikstra** | 197609 | 497291 | 16 | 3.5185 |
| **Greedy (Euclidean)** | - | - | - | - |
| **A\* (Euclidean)** | 181 | 411 | 16 | 0.0030 |

DFS: Not Possible (Infinite Search Tree)

Greedy (Euclidean): My poor laptop (4.6Ghz with 16gb ram) cant handle it </3

Custom Board:

[ "P--------%",

"%-%%%-%%-%",

"%---%----%",

"%-%-%%%-%%",

"%-%------%",

"%----%%%--",

"%----%---."]

**Farmer Puzzle:**

Initial state always set to [False,False,False,False], with goal state always [True,True,True,True].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm (+Heuristic)** | **Nodes Expanded** | **Nodes Generated** | **Solution Cost** | **Time Taken (sec)** |
| **DFS** | - | - | - | - |
| **BFS** | 97 | 210 | 8 | 0.000389337540 |
| **IDA** | 186 | 180 | 8 | 0.000000000000 |
| **Djikstra** | 157 | 342 | 8 | 0.000988245010 |
| **Greedy (Hamming)** | 11 | 22 | 8 | 0.000000000000 |
| **A\* (Hamming)** | 36 | 74 | 8 | 0.000000000000 |

DFS: Not Possible (Infinite Search Tree)

Clearly, Greedy search performs the best, with the least number of nodes expanded (11), followed by A\* with 36 nodes expanded. These two algorithms outperform the others because of the use of the heuristic provided. Indicating that the heuristic is very good. In fact, we can conclude that the heuristic is more useful than the uniform cost assigned to each action, since Greedy algorithm (that only uses heuristics) out performs the rest. Although, we do notice that IDA is very quick (as fast as Greedy and A\*), it expands the most number of nodes, which means the search itself is not very efficient.

Stone Puzzle

TSP: