# Data

Logo Map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Run Time (ms) | Route Distance | Steps Taken | Route Image |
| BFS | 1067.014 | 1709.03334 | 8 |  |
| A\* | 198.989 | 1281.56431 | 4 |  |
| DFS | 213.005 | 1302.96308 | 6 |  |
| Best-First | 319.005 | 1281.56431 | 4 |  |

Game Map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Run Time (ms) | Route Distance | Steps Taken | Route Image |
| BFS | 33440.976 | 1348.57416 | 26 |  |
| A\* | 12415.998 | 1086.59433 | 18 |  |
| DFS | 5023.006 | 1086.59433 | 18 |  |
| Best-First | 32816.999 | 1086.59433 | 18 |  |

Custom Map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Run Time (ms) | Route Distance | Steps Taken | Route Image |
| BFS | 258.988 | 917.94595 | 6 |  |
| A\* | 117.011 | 698.37963 | 6 |  |
| DFS | 136.002 | 698.37963 | 6 |  |
| Best-First | 121.010 | 698.37963 | 12 |  |

# Conclusions

* Logo Map
  + BFS underperforms as is typical, but the simplicity of this map makes any option viable, but with the lowest run time and least steps taken, the edge goes to A\*.
* Game Map
  + In opposition to the last map, the complexity here means that most algorithms end up finding the same route around the island. BFS is bad again, and DFS gets the edge as its simpler construction runs much faster when compared to the other two algorithms with the same path.
* Custom Map
  + The random polygons provide a slightly unique challenge, in that there is a quick route around the edge, but an algorithm that chooses to go through any of the polygons can get stuck and increase cost substantially. A\* wins again but is in a virtual tie with DFS as they found the same route in comparable time. BFS is bad again. Perhaps most interesting is that best-first took as long to complete its path as A\* and DFS, but its greedy approach made in jump inside the mess of polygons, forcing it to bounce around a bit before popping out and resuming the path those other two algorithms took.
* BFS
  + It is naïve, can take a long time, and never outperformed another contender.
* A\*
  + One of the more promising algorithms. It consistently performs well, even best, but can occasionally take a while to complete with more complicated maps. For anything simple or if time is not a concern, A\* is probably the best choice.
* DFS
  + DFS is interesting, as it runs faster than A\* and Best-First in most cases and performs just as well if not better overall. All things considered, if time and complexity are of concern, DFS is a very defensible choice.
* Best-First
  + Highs and lows here. Occasionally finds a near optimal path. Also, occasionally gets diverted and adds steps and distance unnecessarily. Overall, I’m sure there is a better use case than we had in this set of examples, but I wouldn’t choose it over A\* or DFS for any of these problems.