

Comparing Search Algorithms on the Snake Game

Ben Greenfield & Ben Placzek



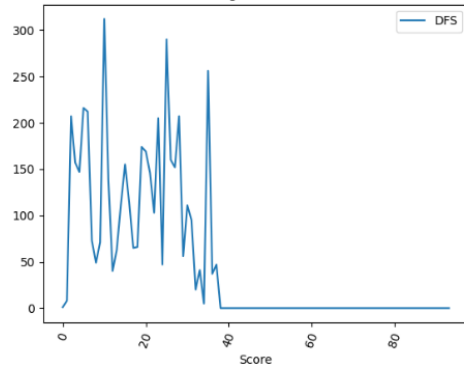
Project Overview

- We looked at the differences in efficiency and success for playing the game of Snake using DFS, BFS, A*, and UCS
- Snake starts as an open grid and slowly gets filled up with your tail (we can treat the squares with our tail as walls).
- We implemented each search algorithm
- We ran a series of trials of each searching algorithm and generated data
- Outputs: Bar graphs, text file data and line graphs

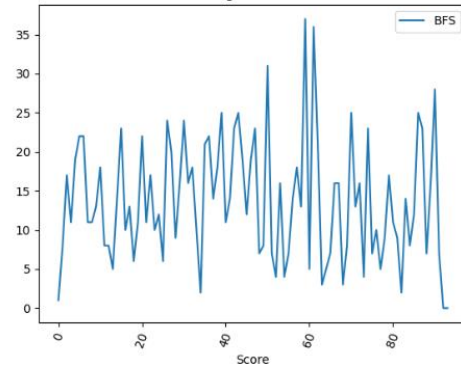
Code Summary

- Snake.py:
 - Game code (Inspired from online resources)
 - Search algorithms (Written by us)
 - runSearch function (snake eats food until dies)
 - Pandas/Dataframe code

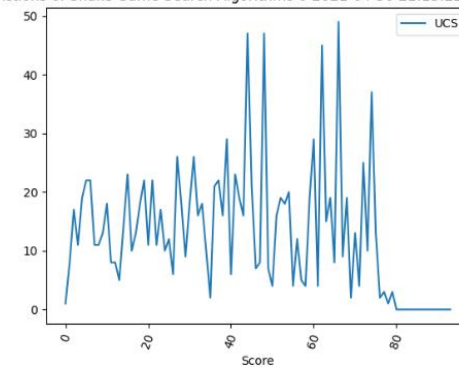
Actions of Snake Game Search Algorithms 0 2021-04-30 21:15:22.211133



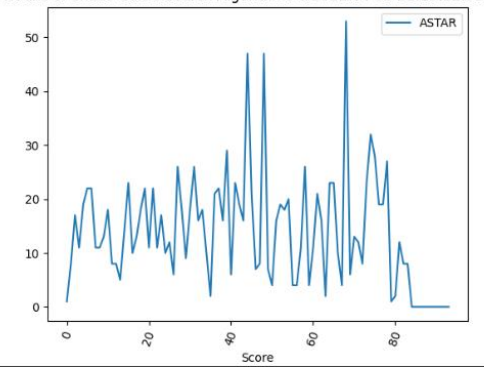
Actions of Snake Game Search Algorithms 0 2021-04-30 21:15:22.346807



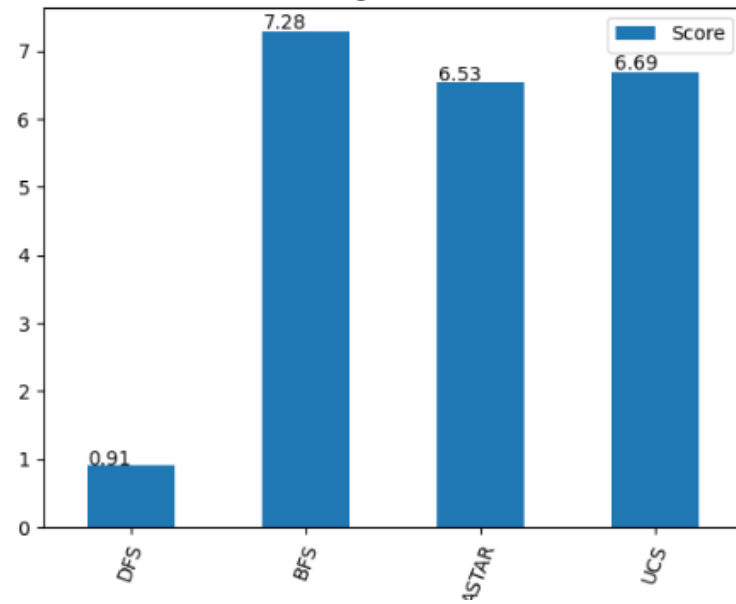
Actions of Snake Game Search Algorithms 0 2021-04-30 21:15:22.634032



Actions of Snake Game Search Algorithms 0 2021-04-30 21:15:22.508852



Scores of Snake Game Search Algorithms 2021-04-30 21:15:22.031618

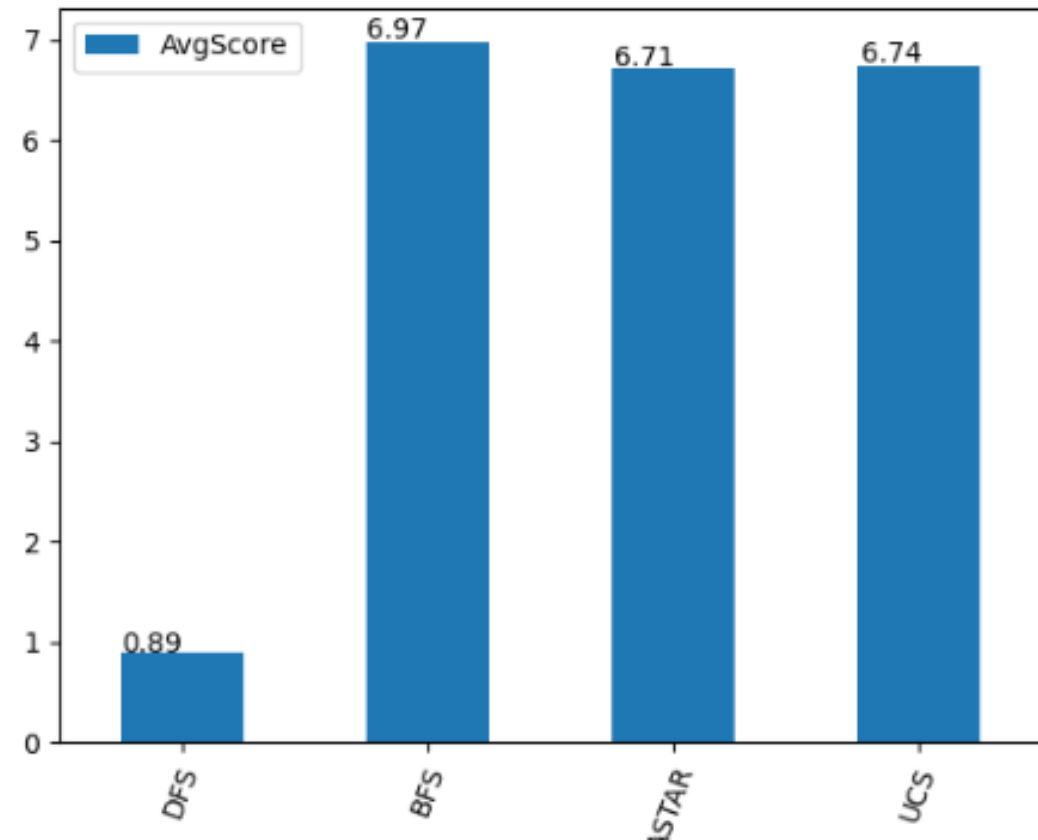


A sample run

Calculated Score

- Calculated Score = $\left(\frac{\text{Score (Number of food eaten)}}{\text{Sum of all actions}} \right) * 100$
- To the right is data based on 5 runs total

Scores of Snake Game Search Algorithms Avg 2021-04-30 21:15:59.472653



Comparing Search Algorithms

- Overall, BFS performs the best
- Summary:
 - DFS: runs slowly with lots of actions, always lowest food score
 - BFS: runs quickly and linearly, good food score, low actions
 - Astar: runs quickly and diagonally, good food score, more actions
 - UCS: runs quickly and diagonally, good food score, more actions

Sources

- <https://www.youtube.com/watch?v=CD4qAhfFuLo&t=1734s>
- https://pastebin.com/embed_js/jB6k06hG