

Maheen Hannan

IN3063/INM702: Mathematics and Programming for AI

Task 1

<https://github.com/MaheenHannan-acvt712/IN3063/tree/3d17fd04cb1a525156047dd114ac2bb274cb3a47/Task1>

Sources used:

My own code from the previous year (98%)

<https://bradfieldcs.com/algos/graphs/dijkstras-algorithm/>

(1%) used heap idea

<https://benalexkeen.com/implementing-dijkstras-shortest-path-algorithm-with-python/>

(1%) used stack reversal method

The basic algorithm operates by directing the agent to the farthest right or bottom cell – the decision is reached through a calculation. The calculation involves either assessing the value to determine the cell with the shortest route or evaluating the absolute difference between the potential destination cell and the current cell. Once the agent reaches an absolute edge, it proceeds directly towards the target cell without further computation. Dijkstra's algorithm transforms the grid into a graph with weighted edges, and the agent traverses the entire grid while assigning the shortest paths to each node. Employing a priority queue where each node has an initial shortest path value of "infinity," the priority queue allows the agent to discard costly paths and enqueue a cheaper path if found. The computation to obtain the shortest path from source to target mirrors the basic algorithm .

The average shortest path for both game modes experiences a logarithmic increase relative to the grid size traversed. This is depicted in "Figure 1, Cost relative to size," where sizes ranging from 0 to 0.1M exhibit a curved incline followed by more linear growth. Consequently, as the grid dimensions expand, so does the travel cost. Furthermore, a growing disparity between the two game modes is observed as the grid size increases, signifying that game mode 2 entails fewer expenses compared to game mode 1. This correlation is reinforced by "Figure 1, Average Cost per Game Mode," as the mean cost disparity between the game modes is approximately ~75%.

However, while game mode 2 boasts a lower average cost, it necessitates significantly greater computational time. This discrepancy is demonstrated in "Figure 1, Average Completion Time per Game Mode," illustrating an approximately 90% disparity in completion time between game mode 1 and 2. Thus, the incorporation of a more intricate cost analysis within the algorithms leads to a substantial augmentation in the time taken for the agent to traverse from source to target.

Furthermore, the time required for game mode 2 exceeds that of game mode 1 by more than 280% at its peak. "Figure 1, Comparison of Time and Size" presents this divergence, wherein at a grid size of 1M, the time needed for game mode 1 to finalize the computation is exponentially lesser than that of game mode 2. Consequently, with increasing grid dimensions, the average shortest path for game mode 2 is reduced by around 90%, accompanied by an almost threefold rise in time.

The efficiency of the game modes is distinctly evident: for expedited computation, game mode 1 holds an advantage, while game mode 2 excels in achieving the shortest path. This dichotomy is aptly illustrated in "Figure 1, Comparison of Cost and Time," highlighting that as costs escalate, so does time, yet the former mode surpasses the latter mode substantially, attaining a level of time efficiency unreached by game mode 1.

The Basic algorithm yields comparable outcomes across both game modes. This is evident from "Figure 2, Division of Computation Time," wherein game mode 2 only marginally surpasses game mode 1, accounting for 1.5% of the overall time. Consequently, within a given threshold, no pronounced preference is discernible concerning the algorithm's selection based on time complexity.

Nonetheless, this 1.5% margin is not mirrored in the agent's shortest path from source to target. "Figure 2, Cost Share Comparison" portrays that game mode 1 predominantly dominates the total cost, exceeding game mode 2 by 8.3%. Utilising the basic algorithm for game mode 1 yields speeds that are 1.5% quicker while incurring an 8.3% increase in cost compared to game mode 2.

Interestingly, the time behaviour of the basic algorithm fluctuates as the grid dimensions expand. Both figures in "Figure 2, Variation in Cost with Time Analysis over Size" (where light blue represents game mode 1 and dark blue represents game mode 2) exhibit a robust

logarithmic trendline: as the grid size expands, the shortest path logarithmically increases. However, the time analysis illustrates that when employed for game mode 1, the algorithm's time for completing the grid wavers with the path increase. Conversely, in the case of game mode 2, the time to finalize the grid remains unaffected by the escalation in the path.

Alternatively, the time analysis focused solely on grid dimensions exposes greater volatility in game mode 2. This is evident in "Figure 2, Time Taken to Complete Grid Size x^2 ," where game mode 2 displays a more pronounced exponential rise in time with heightened variance in contrast to the linear trendline observed for game mode 1, which exhibits a more consistent absence of time variance.

Dijkstra's algorithm demonstrates a marked divergence in computation time between the two game modes. This is illustrated in "Figure 3, Allocation of Computation Time," whereby game mode 1 claims a third of the total computation time, while game mode 2 accounts for the remaining two-thirds. Consequently, for faster computation, game mode 1 emerges as the preferable choice in most scenarios.

However, the approximately 100% augmentation in time for game mode 2 does not linearly correspond to the shortest path. "Figure 3, Comparative Cost Allocation" showcases that, despite often being twice as slow, utilizing Dijkstra's algorithm for game mode 2 merely results in an 8.03% reduction in cost. Intuitively, Dijkstra's algorithm generates longer shortest paths in game mode 1. "Figure 2, Variation in Cost with Time Analysis over Size" reveals that game mode 1 yields greater costs as the grid dimensions expand, exhibiting a more pronounced trend, whereas game mode 2 yields shorter shortest paths with a weaker trend. Consequently, Dijkstra's algorithm, as applied to game mode 1, showcases greater consistency in both time and shortest path outcomes across varying grid sizes.

This consistency in game mode 1's behaviour persists when examining time across grid dimensions. As depicted in "Figure 2, Time Taken to Complete Grid Size x^2 ," game mode 1 demonstrates a linear rise, while game mode 2 manifests an almost exponential escalation. However, this disparity in trend does not translate to time variance, as both modes exhibit similar degrees of variance, although game mode 2 features more notable inconsistencies in higher peaks.

A comparison between the Basic algorithm and Dijkstra's algorithm reveals the following distinctions:

- The Basic algorithm offers swifter computation.
- The Basic algorithm distributes computation time more evenly across diverse cost analyses.
- The Basic algorithm outperforms in terms of per-second performance within a designated grid size limit.
- The Basic algorithm exhibits tighter trendlines for time-related costs.
- The Basic algorithm involves a smaller stack size due to traversing less of the grid.
- Dijkstra's algorithm maintains greater temporal consistency.
- Dijkstra's algorithm consistently produces lower costs in any scenario.
- Dijkstra's algorithm yields notably higher computation time for game mode 2.
- Dijkstra's algorithm exhibits noticeable behavioural differences between game modes.
- Dijkstra's algorithm involves greater spatial and temporal complexities.
- The variance in the shortest path between algorithms is approximately 0.27%.
- The variance in computation time between algorithms for game mode 1 is approximately 14%.
- The variance in computation time between game modes is reduced by approximately 1%.

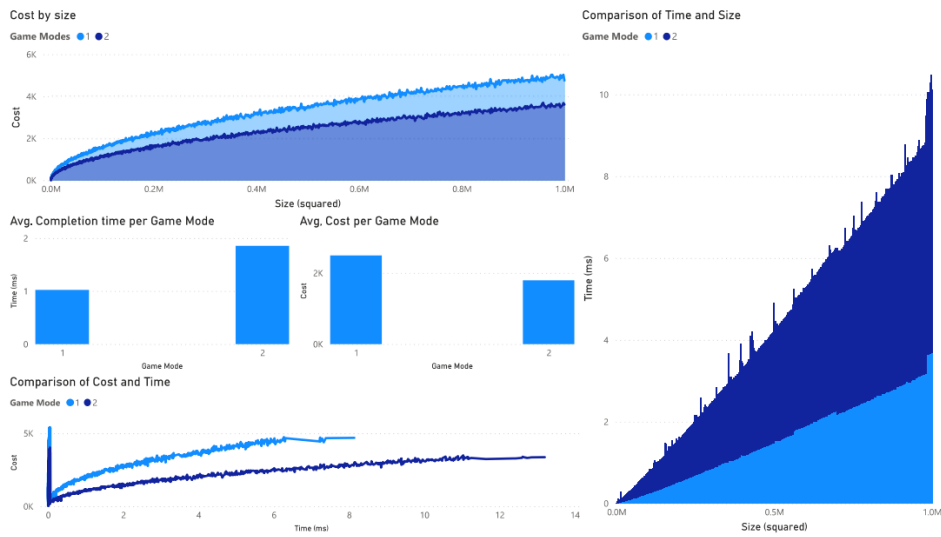


Figure 1 Comparison of Game Mode specific results

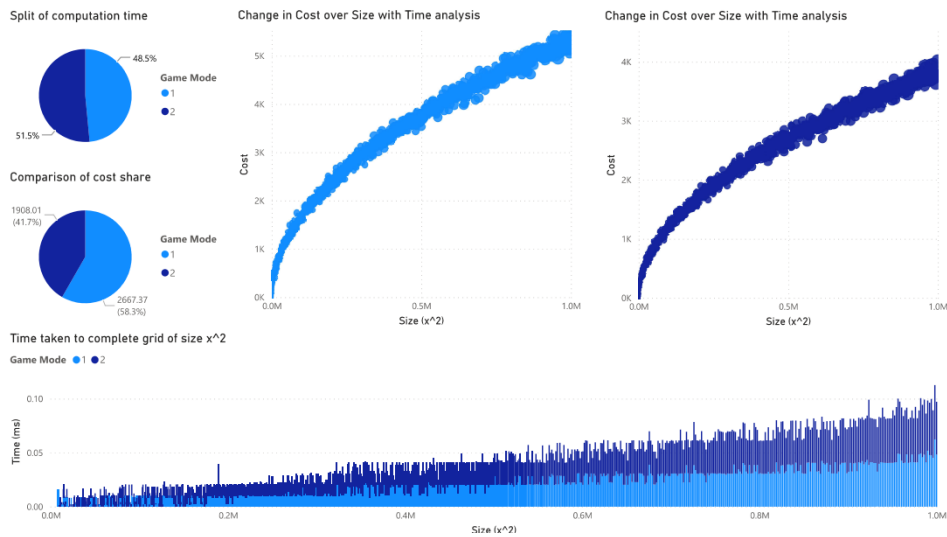


Figure 2 Comparison of Baseline Algorithm results

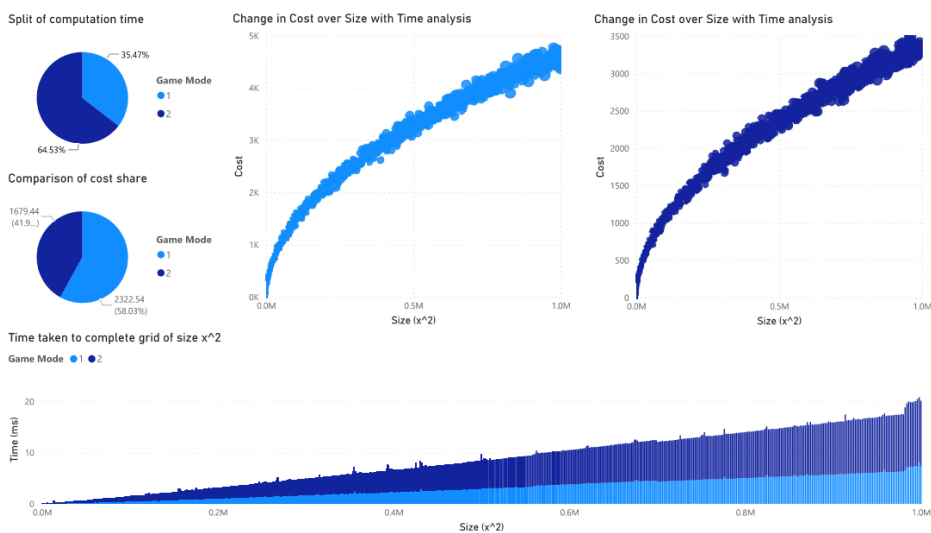


Figure 3 Comparison of Dijkstra's Algorithm results