# Subject: Advanced Intelligence Systems

# Assignment: Program 02 Part 02

# By Saurabh Jawahar Kakade

# Sk2354@nau.edu

# Analysis Answers:

1. Solution:
   1. Searcher class:
      1. Searcher class approach has been taken into consideration.
      2. It has input parameter like filename from graph.
      3. Open function is called for reading graph file.
   2. Breadth First Search Algorithm function:
      1. With the help of list, open nodes and closed nodes are stored.
      2. After initialization of start node, list for open node is appended with start node.
      3. Once the list for open nodes is full (or not empty), outer element of an array is popped out and stored in list of closed nodes and are marked as visited node.
      4. Constant check is applied if goal node is reached or not.
      5. The visited nodes are stored in a list of paths.
      6. BFS algorithms is executed to search nodes in breadth wise manner and to calculate the path cost.
   3. Depth First Search Algorithm Function:
      1. In this function, the procedure to execute is like the BFS function mentioned above.
      2. The only difference is traversing is done in depth wise manner from root node.
   4. A\* Algorithm function:
      1. A\* algorithm is known for finding the shortest path from start node to goal node in each graph.
      2. The cost of edges is used to find the path cost with heuristic function to execute the algorithm.
      3. Graph and heuristics of the graph for each node are initialized in the main function.
      4. Passing parameters for A\* function is graph, heuristics, start node and goal node.
      5. List for open nodes and closed nodes are used like previous mentioned algorithms.
      6. Start node and goal node are initialized.
      7. A continuous loop is executed for open node to reach to goal node.
      8. If the goal node is found in the open node, then the path to reach goal from start node is calculated.
      9. Also, cross check which path is showing the least path cost.
      10. At the end, the path on the graph is printed.
2. Solution:
   1. In heuristic function, every node of the graph has an estimated heuristic value which implements the path cot to goal node.
   2. The output from heuristic function in the form of dictionary is always underestimated and never over estimated (else it will stop functioning).
   3. This heuristic function always calculates the best possible solution for shortest path.
3. Solution:
   1. For 30node.txt map, consider the following solution:

|  |  |  |  |
| --- | --- | --- | --- |
| Start and Goal Node | Breadth-First Search | Depth-First Search | A\* Search |
| H, N | ['H: 0', 'U: 72', 'J: 250', 'T: 453', 'Y: 518', 'N: 684'] | ['H: 0', 'U: 72', 'A: 409', 'W: 493', 'AE: 577', 'B: 625', 'AA: 716', 'N: 816'] |  |
| AE, J | ['AE: 0', 'J: 266'] | ['AE: 0', 'J: 266'] |  |
| B, Q | ['B: 0', 'AC: 119', 'E: 346', 'Q: 795'] | ['B: 0', 'AA: 91', 'T: 184', 'F: 283', 'X: 479', 'C: 598', 'U: 662', 'A: 999', 'W: 1083', 'L: 1254', 'AD: 1342', 'E: 1442', 'Q: 1891'] |  |
| P, K | ['P: 0', 'M: 85', 'K: 223'] | ['P: 0', 'M: 85', 'K: 223'] |  |
| F, A | ['F: 0', 'J: 64', 'U: 242', 'A: 579'] | ['F: 0', 'J: 64', 'AE: 330', 'W: 414', 'A: 498'] |  |

* 1. For 10test.txt map, consider the following solution

|  |  |  |  |
| --- | --- | --- | --- |
| Start and Goal Node | Breadth-First Search | Depth-First Search | A\* Search |
| A, B | ['A: 0', 'G: 127', 'B: 362'] | ['A: 0', 'G: 127', 'B: 362'] | ['A: 0', 'G: 127', 'B: 362'] |
| G, C | ['G: 0', 'A: 127', 'E: 735', 'C: 1028'] | ['G: 0', 'B: 235', 'I: 744', 'C: 1035'] | ['G: 0', 'A: 127', 'H: 199', 'J: 422', 'L: 615', 'C: 749'] |
| F, B | ['F: 0', 'C: 278', 'I: 569', 'B: 1078'] | ['F: 0', 'J: 127', 'H: 350', 'A: 422', 'G: 549', 'B: 784'] |  |
| G, I | ['G: 0', 'B: 235', 'I: 744'] | ['G: 0', 'B: 235', 'I: 744'] | ['G: 0', 'A: 127', 'H: 199', 'D: 334', 'I: 711'] |
| L, C | ['L: 0', 'C: 134'] | ['L: 0', 'C: 134'] | ['L: 0', 'C: 134'] |

1. Analysis:
   1. Algorithm A\* search performs better than others as it executes the efficient optimal path from source to destination with minimum number of nodes traversed.
   2. There are some similarities in path findings between BFS and A\*.
   3. But no other algorithm performs better than A\*.
   4. DFS algorithm performed almost longest path search as compared to others by traversing a greater number of nodes.
   5. A\* always prints the optimal path and hence it guarantees optimal path.
   6. After A\* algorithm, next best performed algorithm was BFS and last is DFS.

* Heuristic Google Maps Function:
  + Google map takes inputs from user like start point and destination.
  + Heuristic function in Google maps performs to find out shortest available path.
  + This function will guide user to reach its destination with short span of time than the longer ones.
  + Heuristic algorithm is best suitable for Google may other than BFS and DFS.
* Program outputs:

Text

Description automatically generated