SEARCH STRATEGIES ANALYSIS

1. THE WATER JUGS PUZZLE:

The water jugs puzzle states that given a set of jugs with certain capacity, try to get to the final amount of water in the jugs at last state. The actions that are with respect to each jug are fill jug, empty jug and transfer amount from one jug to another.

Result for test_jugs.config:

Initial State: Both the jugs are empty

Final State: (1,0)

ALGORITHM	HEURISTIC	TIME COMPLEXITY	SPACE COMPLEXITY FRONTIER	SPACE COMPLEXITY EXPLORED	COMPLETENESS	OPTIMALITY
BFS	NONE	17	4	15	COMPLETE	YES
DFS	NONE	26	3	-	COMPLETE	NOT OPTIMAL
IDDFS	NONE	153	7	-	COMPLETE	YES
UNICOST	NONE	18	5	15	COMPLETE	YES
GREEDY	LOGICAL HEURISTIC	44	23	22	COMPLETE	NOT OPTIMAL
	JACCARD	37	14	24	COMPLETE	NOT OPTIMAL
A*	LOGICAL HEURISTIC	17	5	14	COMPLETE	YES
	JACCARD	17	5	14	COMPLETE	YES
IDA*	LOGICAL HEURISTIC	156	7	-	COMPLETE	YES
	JACCARD	312	7	-	COMPLETE	YES

HEURISTICS:

The heuristic function used for this puzzle is the logical heuristic and Jaccard similarity.

1. Logical Heuristic: This is based on the logic that if either of the jugs is closer to the final state then return the cost as 0 else return the cost as 1. For Example: If the current state in the jug is (1,11) and your final state is (1,0) then it shows that the state is closer to the goal. The cost is returned as 0. If the state is (4,4) and goal is (1,0) then this shows the state is far from the goal and the cost returned would be 1.
    - Admissibility:
      It is Admissible. As the heuristic value (which 0) is less than the cost (which is 1)
    - Consistency:
      It is consistent as the f(n) value for the successor is more than the f(n) value of the parent.

2. Jaccard Similarity: It measures the similarity between finite sample sets and is defined as the cardinality of the intersection of sets divided by the cardinality of the union of the sample sets. For Example: If the current state in the jug is (1,11) and goal state is (1,0) then the jaccard similarity would return a value of (1/3) i.e. 0.66
    - Admissibility:
      It is Admissible. As the heuristic value (which 0) is less than the cost (which is 1)
    - Consistency:
      It is consistent as the f(n) value for the successor is more than the f(n) value of the parent.

INFERENCE:

- Both the heuristic was found to be admissible and consistent.
- It is seen that Jaccard similarity performed better than the Logical Heuristic as the space complexity is similar and there is a huge difference in the time complexity for all the informed search strategies.
- It was observed that the DFS and Greedy did not provide optimal solutions.
    - DFS: This search strategy did not give an optimal solution as it starts at the root and explores as far as possible along each branch before backtracking.
    - Greedy: Here Greedy did not provide the optimal solution because the heuristic provided was not optimal for this puzzle to run through greedy algorithm.
- It is seen that the space complexity is similar for BFS, Uniform Cost and A*. Here we have the frontier as queue or priority queue and the explored list maintains the number of nodes visited.
- The space complexity for the IDDFS and IDA* is similar. Since the explored list is not used in IDDFS and IDA*, there is a huge difference in the space complexity compared to the other search.
- Although the space complexity is less for the IDDFS and IDA*, the BFS, Uniform Cost and A* seems to perform better as they have less time complexity at greater extent than IDDFS and IDA* search strategies.

2. THE PATH FINDING PUZZLE:

The path finding puzzle aims at finding the path from one place to another. The action would be to find the next close place in order to reach the goal.

Result for test_cities.config:

Initial State: Starting point (C00)

Final State: Destination point (C04)

ALGORITHM	HEURISTIC	TIME COMPLEXITY	SPACE COMPLEXITY FRONTIER	SPACE COMPLEXITY EXPLORED	COMPLETENESS	OPTIMALITY
BFS	NONE	317	181	24	COMPLETE	NOT OPTIMAL*
DFS	NONE	72	51	-	COMPLETE	NOT OPTIMAL*
IDDFS	NONE	198	19	-	COMPLETE	NOT OPTIMAL*
UNICOST	NONE	161	76	24	COMPLETE	YES
GREEDY	MANHATTAN	24	21	4	COMPLETE	NOT OPTIMAL*
	EUCLIDEAN	24	21	4	COMPLETE	NOT OPTIMAL*
A*	MANHATTAN	95	52	24	COMPLETE	YES
	EUCLIDEAN	94	51	24	COMPLETE	YES
IDA*	MANHATTAN	779	30	-	COMPLETE	YES
	EUCLIDEAN	8210	34	-	COMPLETE	YES

*Since the path cost is taken into consideration the optimality cannot be determined until cost is calculated. Based on the output, it doesn't give the path with least cost. (It is not the same as the uniform cost as uniform cost gives an optimal solution.)

HEURISTICS:

The heuristic function used for this puzzle is the Manhattan Distance and Euclidean Distance.

1. Manhattan Distance: It is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates.
    - Admissibility:
      It is Admissible. As the heuristic value at that point is less than the path cost taken to reach that point.
    - Consistency:
      It is consistent as the f(n) value for the successor is more than the f(n) value of the parent.

2. Euclidean Distance: The Euclidean distance between two points is the length of the path connecting them.
    - Admissibility:
      It is Admissible. As the heuristic value at that point is less than the path cost taken to reach that point.
    - Consistency:
      It is consistent as the f(n) value for the successor is more than the f(n) value of the parent.

INFERENCE:

- Both the heuristic was found to be admissible and consistent.
- It is seen that for all informed search strategies, the Euclidean distance and Manhattan Distance performed consistently expect in IDA*. In IDA*, the Euclidean distance's time complexity was way higher than the Manhattan Distance
- It was observed that the BFS, DFS, IDDFS and Greedy did not provide optimal solutions. My definition of optimality for this puzzle is that the path with the least cost in the most optimal solution. The optimal is the uniform cost search.
    - BFS: The reason would be that since cost is not taken into account, the solution is found at an early level at the neighbor nodes and is inconsistent to the uniform cost search.

- DFS: The reason would be that since cost is not taken into account. This search strategy did not give an optimal solution as it starts at the root and explores as far as possible along each branch before backtracking. The solution obtained is inconsistent to the uniform cost search.
- IDDFS: The solution obtained is inconsistent to the uniform cost search.
- Greedy: The solution obtained is inconsistent to the uniform cost search.

- Based on the time complexity and Space complexity analysis, the worst performance was observed in BFS as it takes more time and more memory.
- Although uniform cost, A* and IDA* search strategies gave the optimal output, the best search strategy is the A* as the time complexity and Space complexity is less compared to the other two strategies.

3. THE BURNT PANCAKE PUZZLE:

This puzzle is a variation of the pancake flipping problem. Here the pancakes are sorted in the order of their size and burnt side is faced down.

Result for pancake6.config:

Initial State: Initial State (2, 1, 4, -5, 3)

Final State: Final State (1, 2, 3, 4, 5)

ALGORITHM	HEURISTIC	TIME COMPLEXITY	SPACE COMPLEXITY FRONTIER	SPACE COMPLEXITY EXPLORED	COMPLETENESS	OPTIMALITY
BFS	NONE	5744	3804	1263	COMPLETE	YES
DFS	NONE	6853	4499	-	COMPLETE	NOT OPTIMAL
IDDFS	NONE	8342	20	-	COMPLETE	YES
UNICOST	NONE	8880	5510	1904	COMPLETE	YES
GREEDY	JACCARD	2278	1524	566	COMPLETE	NOT OPTIMAL
A*	JACCARD	4595	3030	1098	COMPLETE	YES
IDA*	JACCARD	35689	20	-	COMPLETE	YES

HEURISTICS:

The heuristic function used for this puzzle is the Jaccard Similarity.

1. Jaccard Similarity: It measures the similarity between finite sample sets and is defined as the cardinality of the intersection of sets divided by the cardinality of the union of the sample sets. For Example: If the current state is (1,11) and goal state is (1,0) then the jaccard similarity would return a value of (1/3) i.e. 0.66
    - Admissibility:
      It is Admissible. As the heuristic value (which 0) is less than the cost (which is 1)
    - Consistency:
      It is consistent as the f(n) value for the successor is more than the f(n) value of the parent.

INFERENCE:

- IDA* performed the worst with respect to the time complexity. Took the maximum time compared to any of the search strategies.
- Uniform Cost performed the worst for this puzzle as it took the maximum time and space complexity combination compared to the other search strategies although it gave optimal solution.
- Based on both the combination of the time complexity, space complexity and optimality, the IDDFS seemed to perform better than others. Then comes A*.
- Since the time complexity and space complexity is huge, the test cases provided to us didn't complete within the time frame allotted to us.