Artificial Intelligence - Report

Devyansh Chaudhary - 2022156

Q3 [Given solutions for problem 3's parts where reasoning is asked]

Part B:

Yes, I found out that the path between two nodes comes out to be the same for Iterative Deepening Search and Bidirectional BFS for the give\\n graph and in given public test cases all the paths matched with each other.

But, No True for all time, This will not always be the true case there can be situations where the paths may change because of the directed graph nature, The BiBSF may sometimes get best path sometimes IDS may get the best possible path with minimum nodes. The IDS may stuck in a long chain of nodes because the Goal Node and Start Node have an edge from Goal to Start, In that case, BiBSF will return the best path as it will get the meeting node as Goal Node only and on path reconstruction, it will return better path then long chained path of IDS.

```
Comparing IDS and BiBFS for nodes 0 -> 47:
Bidirectional BFS Path:  [0, 49, 47]
IDS Path:  []
```

0 ←49←47
For Example in this case, there is a bath from Goal Node(47) to Source Node(0) but not From Source to Goal for that IDS will not be able to find a path whereas bidirectional applying search from goal and search both so BiBSF is able to find a path but not by IDS.

Part C:

```
Below results are for all node pairs
Total Execution Time for IDS: 1468.3593s
Total Execution Time for Bidirectional BFS: 728.5262s
Total memory usage for IDS: 358939.44KB
Total memory usage for Bidirectional BFS: 445330.36KB
```

From this, I can interpret that Even though Bidirectional BFS uses High Memory because of the Queues DS using but gives faster output because of applying BFS from both ends having time complexity $b^{d/2}$ On the other hand, IDS takes Low memory usage as only using stack space for DFS but will take High runtime because of iterative searching of node also using DFS, this involves repetition of search time for nodes already discovered at lower iterations.

Part E:
(b)No, I found out that the paths of A* and Bidirectional Heuristic were not the same for the given public test cases.

But this is not always true that they will have different; they may have the same paths returned by both the algorithms in the given graph found a case for that to in the graph:

```
Comparing algorithms for nodes 1 -> 3:
[1, 27, 9, 2, 11, 32, 31, 3]
[1, 27, 9, 2, 11, 32, 31, 3]
```

As for this node pair, there is no better path from any node (u to w to v) where w comes before the goal node in the resulted path, and this is similarly true at the same time from the goal node side; there is no w from the goal node side, which has a better path to Start Node in the resulted path from the reverse side (v to w to u). Here better path means less cost and cost is sum of the distance of (u, w) and (w, v)

(c)

From the Time Execution and Memory Usage for the algorithms, we can interpret A* and Bidirectional Heuristic search both have taken moderately same time in execution to find nodes, While for cost Bidirectional Heuristic takes more memory cost because of min heap priority queues take more memory cost while in A* there is no extra memory cost then storing heuristic values which were same for Bidirectional Heuristic search therefore this is the reason for Bidirectional for having more memory usage.

```
For all node pairs
Total Execution Time for A*: 820.0568s
Total Execution Time for Bidirectional Heuristic Search: 819.0265s
Total Memory Usage for A*: 435352.42KB
Total Memory Usage for Bidirectional Heuristic Search: 581819.92KB
```

Part F:

Plot 1: This plot is for Execution Time vs Memory Time this compares all 4 algorithms; from the graph, I interpreted that IDS takes the longest time while the memory usage is around 28000KB, the IDS clusters the most because of time is higher in that case stack space is also higher. Bidirectional Heuristics takes the more memory usage high while taking average time only as compared to other algorithms. The A* has performed better in terms of both memory vs runtime, both runtime and memory usage. Bidirectional BFS uses high memory also less memory in some cases it depends on the path length and graph size that connects start and goal nodes it has to store more nodes in queues which increases the memory usage while the Time of Execution is not changes a lot it is almost constants.

Informed Searches: has taken moderately more memory usage but has performed better in terms of execution time.
Uniformed Searches: has taken a lot More High time than uninformed but has performed better than unformed in termes of memory usage. While sometimes Bidirectional Search has taken a comparatively same time of execution sometimes.

Plot 2: Path cost for Bidirectional Heuristic is spread out form 0 to 30000KB while IDS only gives higher cost compared to other algorithms also takes more time than any other algorithms therefore this shows that IDS has performed worst of all the other algorithms. The A*, Bidirectional and Bidirectional Heuristics all have the path cost spread out but on comparing Bidirectional search has also performed worse than informed searches,

Informed Searches: has performed better than Uniformed in case of time complexity vs path cost, the found lesser path cost than uniformed and also in lot lesser time.

Uninformed Searches: have worse than informed because thay used a lot more time like IDS also a higher path cost like Bidirectional BFS.