AI Assignment-1 Report

Objective : The main objective of this report is to provide a brief analysis of the searching algorithms (BFS, DFS, GBFS and A*) over a random connected graph that contains names of the cities. It aims to give insights of these algorithms pros and cons after analysis of the graph.

1. Number of total nodes in complete graph = 45
2. Number of total edges in complete graph = 2025
3. Number of nodes in the random connected graph = 45
4. Number of edges in the random connected graph = 202
5. Dropout chosen to create a random graph from a complete graph = 0.85

These algorithms were run over five pairs of nodes from the random connected graph as mentioned in the assignment : Five pair of nodes are:

1. Patna to Una
2. Mirzapur to Delhi
3. Jaipur to Prayagraj
4. Lucknow to Arrah
5. Lakhimpur to Bikaner

Case 1 : Starting node Patna and Target node Una.

Path	Algorithms	Nodes in path list	Total visited nodes	Time (in sec)	Memory (in bytes)	Comments
Patna To Una	**BFS**	**5**	**29**	**0.7141**	**96**	BFS is taking **maximum space,** visited 29 nodes for finding the path and DFS taking **maximum time** to finding the path. GBFS is **not able to find** the path
	DFS	**7**	**16**	**2.000146**	**112**	
	GBFS	**0**	**0**	**0.00156**	**0**	
	A*	**5**	**5**	**0.00724**	**96**	

Case 2: Starting node Jaipur and Target node Prayagraj

Path	Algorithms	Nodes in path list	Total visited nodes	Time (in sec)	Memory (in bytes)	Comments
Jaipur To Prayagraj	**BFS**	**4**	**32**	**0.000349**	**88**	DFS takes **maximum space** , it has to visited 37 nodes to finding the path and A* takes maximum **time**. GBFS is most effective in this case (**minimal**)
	DFS	**23**	**37**	**0.000244**	**280**	
	GBFS	**5**	**5**	**0.0183**	**152**	
	A*	**4**	**4**	**0.2157**	**88**	

Case 3: Starting node Mirzapur and Target node Delhi

Path	Algorithms	Nodes in path list	Total visited nodes	Time (in sec)	Memory (in bytes)	Comments
Mirzapur To Delhi	**BFS**	**3**	**4**	**1.000246**	**80**	In this case DFS is giving **wrost output**, taking maximum time and space . **A*** is most effective in this case.
	DFS	**22**	**22**	**4.00197**	**280**	
	GBFS	**7**	**7**	**0.02321**	**112**	
	A*	**3**	**3**	**0.0079**	**80**	

Case 4 : Starting node Lucknow and Target Node Arrah.

Path	Algorithms	Nodes in path list	Total visited nodes	Time (in sec)	Memory (in bytes)	Comments
Lucknow To Arrah	**BFS**	**3**	**7**	**0.00336**	**80**	Although BFS give **optimal** solution but is has high space complexity also. Solution of GBFS is both **optimal and minimal**
	DFS	**4**	**4**	**1.00244**	**88**	
	GBFS	**3**	**3**	**0.01193**	**80**	
	A*	**5**	**5**	**0.0438**	**96**	

Case 5 : Starting Node Lakhimpur and Target Node Bikaner

Path	Algorithms	Nodes in path list	Total visited nodes	Time (in sec)	Memory (in bytes)	Comments
Lakhimpur To Bikaner	BFS	7	36	2.0087	112	**DFS** is taking maximum time and space. GBFS is **not able to find** out the path at all.
	DFS	25	41	5.000244	312	
	GBFS	0	0	0.00063	0	
	A*	7	7	0.00835	112	

The Average Ananlysis :

Experiment	Algorithms	Average Time (in s)	Space (no. of nodes stored in path list)	Observation
Average	BFS	0.7488	91.2	If we see on an average DFS takes maximum space as well as maximum time in order to finding the path.
	DFS	2.448	214.4	
	GBFS	0.1124	68.8	
	A*	0.0566	94.4	

Detailed Explanation:

1. **Analysis of Breadth First Search:**
   As we know the BFS algorithm is lies in uninformed search algorithms and It is optimal means it will surely give path if the nodes exists in the graph. We able to find the shortest path.

Pros of BFS:

It is optimal and give the shortest path
In case 3 (Jaipur to Prayagraj) :
Here the bfs gave the very optimal solution that it's path list contain only 3 nodes and the total no. of nodes visited is 4. As compared to other algorithms like dfs : it path list has 22 nodes as well it has to traverse through 22 nodes in this case bfs very helpful. Hence bfs gave the shortest path.

In case 5(Lakhimpur to Bikaner):
Path list contain 4 nodes while total of nodes visited 36. But if we see dfs Itpath list contain 25 and total no. of nodes visited is 41 Hence it has less space complexity this specific case

Cons of BFS:
In case 1 : (Patna to Una) :

The path list contains 5 nodes , while it has to visit total 29 nodes in order to reach it's goal node (Una). Basically bfs explores level by level that's why it has to open so many nodes as compared to dfs which has to explores only 7 nodes.
Here we have to worry about space complexity.

2. **Analysis of Depth First Search :**
   DFS is algorithm is also lies under uninformed search algorithms . It traverse the graph depth wise and then start back tracking in order to reach it's goal.

   Pros of DFS:
   As it traverse depth wise that's the reason it has efficient memory management as compared to BFS.
   In case 4: (Lucknow to Arrah)
   Path list contains 4 nodes as well as total visited nodes is also 4 .
   Here it finds the solution in less space and time complexity as compared to bfs.

Cons of DFS:

In case 5: (Lakhimpur to Bikaner)

It's path list contains 25 nodes and it has to visited 41 nodes that is very expensive and does not give the shortest path as well as compared to BFS.

3. **Analysis of Greedy Best First Search:**

   GBFS algorithm is informed search algorithm. It is used to final minimal solutions but it is not optimal . It explores the nodes in a specific direction according to the heuristic value of that node . It is admissible but there is no backtracking it moves according to heuristic value.
   It may or may not find the solution and stuck at dead end.
   Pros :
   It always give the fastest and shortest solution in less time because it moves only in one direction means no backtracking.

   In case : 2 (Mirzapur to Delhi) and In case 2 (Jaipur to Prayagraj)
   It is giving the shortest path same as BFS Hence it is giving the minimal as well as optimal solution in this case
   In GBFS , It will visited only those nodes which has less heuristic value.
   It means it only visited no. of nodes that are in it's path list

   Cons:
   Since GBFS is minimal but there is an issue that it may have chance that is stuck at the dead end without giving the solution.
   Although I tried but unfortunately I didn't find any dead node in my random connected graph.

   In case our Heuristic Function is not properly defined than GBFS we didn't get the minimal solution. There is ambiguity in the solution.

4. **Analysis of A* Algorithm:**

   A* algorithm is a informed search algorithm. It explores the nodes in a specific direction and with a effective manner. The expand of next node
   Is depend on the value of f(n) which is sum of g(n) and h(n) .

   f(n) = g(n) + h(n)

Where **g(n**) function gives the route distance between start node and goal node

h(n) function give the heuristic distance between start node and goal node.

Pros:

A* algorithm is optimal. It will surely give the shortest between two nodes if they exists in the graph.
It is more effective than GBFS because it has backtracking feature. It can backtrack in order to give optimal solution.
Due to backtracking it will never stuck at a dead end as compared to GBFS.
In case 5 (Lakhimpur to Bikaner)
It give optimal and minimal solution than GBFS because it has g(n) and there is backtracking helped to get shortest path.

Cons:
Since A* algorithms is optimal but it has issues that for different problems have different heuristic functions defined . In that case the performance of A* will vary respectively. it is optimal but it is not minimal. Hence the traversal is very expensive due backtracking feature. It also have high space and time complexities as compared to other algorithms.
When A* is applied on large data set it is very expensive to find optimal solution.

Note : I have used internet in order to debug or implement GBFS and A* Algorithm.

Conclusion : All the above algorithms have some cons and pros . But the use of these algorithms based on the problems rather than applying them randomly. Firstly we have to identify that which algorithm would be best fit in order to get the optimal or desired solution.

Author: Anmol Kumar