Methods	Expansions				Time (seconds)				Plan length			
	P.1	P.2	P.3	P.4	P.1	P.2	P.3	P.4	P.1	P.2	P.3	P.4
Breadth First Search	43	3343	14663	99736	0.006	1.808	9.568	89.44	6	9	12	14
Depth First Graph Search	21	624	408	25174	0.003	2.734	1.075	3654.2	20	619	392	24132
Uniform Cost Search	60	5154	18510	113339	0.008	3.074	13.237	157.41	6	9	12	14
Greedy Best First Graph Search with h_unmet_goals	7	17	25	29	0.001	0.017	0.0333	0.1179	6	9	15	18
Greedy Best First Graph Search with h_pg_levelsum	6	9	14	17	0.397	8.963	19.912	55.596	6	9	14	17
Greedy Best First Graph Search with h_pg_maxlevel	6	27	21	56	0.2966	17.998	24.308	133.89	6	9	13	17
Greedy Best First Graph Search with h_pg_setlevel	6	9	35	107	0.545	13.159	73.213	512.80	6	9	17	23
Astar search with h_unmet_goals	50	2467	7388	34332	0.008	2.048	8.121	61.149	6	9	12	14
Astar search with h_pg_levelsum	28	357	369	1208	1.014	228.51	579.93	3326.9	6	9	12	15
Astar search with h_pg_maxlevel	43	2887	9580	62077	1.0602	1311.7	9998.4	18535.2	6	9	12	14
Astar search with h_pg_setlevel	33	1037	3423	37912	1.2994	1190.1	9924.9	54569.5	6	9	12	14

Table illustrating Expansion values, Time in seconds and Plan length for all 4 problems for different algorithms

Methods	Expansions				Actions			
	P.1	P.2	P.3	P.4	P.1	P.2	P.3	P.4
Breadth First Search	43	3343	14663	99736	20	72	88	104
Depth First Graph Search	21	624	408	25174	20	72	88	104
Uniform Cost Search	60	5154	18510	113339	20	72	88	104
Greedy Best First Graph Search with h_unmet_goals	7	17	25	29	20	72	88	104
Greedy Best First Graph Search with h_pg_levelsum	6	9	14	17	20	72	88	104
Greedy Best First Graph Search with h_pg_maxlevel	6	27	21	56	20	72	88	104
Greedy Best First Graph Search with h_pg_setlevel	6	9	35	107	20	72	88	104
Astar search with h_unmet_goals	50	2467	7388	34332	20	72	88	104
Astar search with h_pg_levelsum	28	357	369	1208	20	72	88	104
Astar search with h_pg_maxlevel	43	2887	9580	62077	20	72	88	104
Astar search with h_pg_setlevel	33	1037	3423	37912	20	72	88	104

Table analyzing Expansion values vs Actions values for all 4 problems for different search algorithms

Methods	Time (seconds)				Actions			
	P.1	P.2	P.3	P.4	P.1	P.2	P.3	P.4
Breadth First Search	0.006	1.808	9.568	89.44	20	72	88	104
Depth First Graph Search	0.003	2.734	1.075	3654.2	20	72	88	104
Uniform Cost Search	0.008	3.074	13.237	157.41	20	72	88	104
Greedy Best First Graph Search with h_unmet_goals	0.001	0.017	0.0333	0.1179	20	72	88	104
Greedy Best First Graph Search with h_pg_levelsum	0.397	8.963	19.912	55.596	20	72	88	104
Greedy Best First Graph Search with h_pg_maxlevel	0.2966	17.998	24.308	133.89	20	72	88	104
Greedy Best First Graph Search with h_pg_setlevel	0.545	13.159	73.213	512.80	20	72	88	104
Astar search with h_unmet_goals	0.008	2.048	8.121	61.149	20	72	88	104
Astar search with h_pg_levelsum	1.014	228.51	579.93	3326.9	20	72	88	104
Astar search with h_pg_maxlevel	1.0602	1311.7	9998.4	18535.2	20	72	88	104
Astar search with h_pg_setlevel	1.2994	1190.1	9924.9	54569.5	20	72	88	104

Table analyzing Times in seconds' vs Actions values for all 4 problems for different search algorithms

- Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?
- Uniformed search algorithms like Breadth First Search, Depth First Search, and Uniform Cost Search will be appropriate for planning in a very restricted domain as they outperform all other search algorithms in terms of time complexity for problem 1(problem 1 has very few actions).

- Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)?
- Greedy Best First Graph Search will be appropriate for this situation as they have better time complexity than A star algorithms and perform well on large domains as evident from results for problem 2 and 3. A star algorithms are also well suited; however, they are less efficient with respect to the time complexity.

- Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?
- A star algorithms will prove to be the best of all if the sole aim is to find the optimal paths. As can be viewed from the Expansion values, the number of expansions by A star algorithms is much larger than other peer algorithms. This ensures that A star is more likely to find the optimal paths.