Introduction:

This report is for the Artificial Intelligence Nanodegree planning project. It is split into metrics for each problem, Analysis, and answers for questions subtitled Discussions.

Metrics:

Problem 1: Actions = 20

Algorithm	Expansions	Search Time	Length of Plan
BFS	43	.006	6
DFS	21	.003	20
UCS	60	.01	6
GBFS unmet	7	.002	6
GBFS levelsum	6	.44	6
GBFS maxlevel	6	.33	6
GBFS setlevel	6	.52	6
A* unmet	50	.01	6
A* levelsum	28	1.13	6
A* levelmax	43	1.17	6
A* levelset	33	1.21	6

Problem 2: Actions = 72

Algorithm	Expansions	Search Time	Length of Plan
BFS	3343	2.03	9
DFS	624	3.14	619
UCS	5154	3.39	9
GBFS unmet	17	.02	9
GBFS levelsum	9	10.4	9
GBFS maxlevel	27	21	9
GBFS setlevel	9	13.05	9

A* unmet	2467	2.3	9
A* levelsum	357	265	9
A* levelmax	2887	1523	9
A* levelset	1037	1168	9

Problem 3: Actions = 88

Algorithm	Expansions	Search Time	Length of Plan
BFS	14663	11.24	12
GBFS unmet	25	.04	15
GBFS levelsum	14	23.55	14
A* unmet	7388	8.75	12
A* levelsum	369	428	12

Problem 4: Actions = 104

Algorithm	Expansions	Search Time	Length of Plan
BFS	99736	100	14
GBFS unmet	29	.06	18
GBFS levelsum	17	42.75	17
A* unmet	34330	57.17	14
A* levelsum	1208	2420	15

Analysis:

Analysis of search complexity:
For uninformed searches such as Breadth First Search, as problem action space expands, the number of expansions is increasing exponentially. Greedy Best First Search has a constant low number of expansions regardless of domain increase. For A*, we see the number of expansions increase, however, level sum heuristic produces less expansions than unmet goals heuristic.

Analysis of search time:
For restricted problems, surprisingly, informed search algorithms produces better results than A*. However, as action space increases, we see that search time expands for these problems.

However, they still produce decent results compared to A* with level sum heuristic.This could be due to inefficient implementation of level sum heuristic in my code as A* and GBFS with unmet goals produces good results compared to BFS.

Analysis of search optimality:
Search optimality is the same for all algorithms if the problem is restricted except for Depth First Search which is very inefficient in this regards. Search optimality varies when a problem becomes more complex but they average the same.

Discussion:

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?

In this case, the best algorithm according to problem 1 analysis which is very restricted, is Greedy Best First Search with unmet goals heuristic as it gives almost instantaneous results. Along with that, uninformed searches such as Breadth First Search and Depth first search gives good results too.

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)

In this case, we do not care about time efficiency but rather optimality of solution. All algorithms would produce an optimal solution, however, some of them are more complex than others. The least complex algorithm is Greedy Best First Search with all of its heuristics.

Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

Greedy Best First Search as number of expansions is at minimum with thi algorithm.