	Air Cargo Problem 1	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Time (pypy3)
	uninformed search						
1	breadth_first_search	20	43	56	178	6	0.023811796
2	depth_first_graph_search	20	21	22	84	20	0.009618925
3	uniform_cost_search	20	60	62	240	6	0.017927114
	greedy_best_first_graph_search						
4	h_unmet_goals	20	7	9	29	6	0.001633515
5	h_pg_levelsum	20	6	8	28	6	0.730893023
6	h_pg_maxlevel	20	6	8	24	6	0.117956296
7	h_pg_setlevel	20	6	8	28	6	0.882086453
	astar_search						
8	h_unmet_goals	20	50	52	206	6	0.011246378
9	h_pg_levelsum	20	28	30	122	6	0.534907814
10	h_pg_maxlevel	20	43	45	180	6	0.261162691
11	h_pg_setlevel	20	33	35	138	6	1.3706316

	Air Cargo Problem 2	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Time (pypy3)
	uninformed search						
1	breadth_first_search	72	3343	4609	30503	9	0.337785426
2	depth_first_graph_search	72	624	625	5602	619	0.496396211
3	uniform_cost_search	72	5154	5156	46618	9	0.571888791
	greedy_best_first_graph_search						
4	h_unmet_goals	72	17	19	170	9	0.009136501
5	h_pg_levelsum	72	9	11	86	9	3.692731617
6	h_pg_maxlevel	72	27	29	249	9	2.649279092
7	h_pg_setlevel	72	9	11	84	9	11.851160706
	astar_search						
8	h_unmet_goals	72	2467	2469	22522	9	0.673549578
9	h_pg_levelsum	72	357	359	3426	9	106.94428423
10	h_pg_maxlevel	72	2887	2889	26594	9	250.266466691
11	h_pg_setlevel	72	1037	1039	9605	9	890.48271614

	Air Cargo Problem 3	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Time (pypy3)
	uninformed search						
1	breadth_first_search	88	14663	18098	129625	12	0.947339138
2	depth_first_graph_search	88	408	409	3364	392	0.257007012
3	uniform_cost_search	88	18510	18512	161936	12	1.413229058
	greedy_best_first_graph_search						
4	h_unmet_goals	88	25	27	230	15	0.020584356
5	h_pg_levelsum	88	14	16	126	14	11.83760229
6	h_pg_maxlevel	88	21	23	195	13	3.709348562
7	h_pg_setlevel	88	35	37	345	17	57.184571421
	astar_search						
8	h_unmet_goals	88	7388	7390	65711	12	1.376859467
9	h_pg_levelsum	88	369	371	3403	12	156.706902662
10	h_pg_maxlevel	88	9580	9582	86312	12	1259.335336112
11	h_pg_setlevel	88	3423	3425	31596	12	4236.97163035

	Air Cargo Problem 4	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Time (pypy3)	
	uninformed search							
1	breadth_first_search	104	99736	114953	944130	14	6.001156547	
2	depth_first_graph_search	104	25174	25175	228849	24132	1112.628073653	
3	uniform_cost_search	104	113339	113341	1066413	14	11.360931355	
	greedy_best_first_graph_search							
4	h_unmet_goals	104	29	31	280	18	0.044849814	
5	h_pg_levelsum	104	17	19	165	17	14.502743878	
6	h_pg_maxlevel	104	56	58	580	17	10.10761648	
7	h_pg_setlevel	104	107	109	1164	23	232.459051012	
	astar_search							
8	h_unmet_goals	104	34330	34332	328509	14	4.113155945	
9	h_pg_levelsum	104	1208	1210	12210	15	776.447068099	
10	h_pg_maxlevel	104	62077	62079	599376	14	11914.10174954	Over 3 hours
11	h_pg_setlevel	104	22606	22608	224229	14	39854.09688239	11 hours overnight

Bar charts and Pie charts in notebook ‘Udacity AIND Project 2 Planning - Results Data Analysis.ipynb’

Use your results to answer the following questions:

Q1. 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?

Answer: None

Looking at the results, the number of actions are all the same for all search algorithms in each problem.
The number of actions only differ by different problems, not by difference in algorithms.
However, Air Cargo Problem 1 results have the lowest number of actions of 20,
the fastest search algorithm for air cargo problem 1 is 'greedy_best_first_graph_search' with 'h_unmet_goals',
with run time of only 0.001634 seconds in my local environment.

Q2. 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)

Answer: uniform cost search

I assume very large domain needs large number of expansions with minimum of time.
Looking at the data of all four air cargo problems, the 'uniform cost search' search algorithm always have the highest numbers of expansions.
The time taken running 'uniform cost search' using pypy is very small, not the smallest, but definitely one of the quickest.

The number of expansions and time of the 'uniform cost search' algorithm in each problem:
In Air Cargo Problem 1, expansion is 60, in 0.0179 second
In Air Cargo Problem 2, expansion is 5154, in 0.5718 second
In Air Cargo Problem 3, expansion is 18510, in 1.4132 second
In Air Cargo Problem 4, expansion is 113339, in 11.3609 second

Q3. Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

Answer:

I assume lowest plan length is best for the optimal plan.
From data, these are the 5 search algorithms best for optimal plans with the lowest total plan length 41 (see table below)

breadth first search
uniform cost search
astar with 'h unmet goals'
astar with 'h pg maxlevel'
astar with 'h pg'setlevel'

search algorithm	Problem 1	Problem 2	Problem 3	Problem 4	Total Plan Length
breadth_first_search	6	9	12	14	41
depth_first_graph_search	20	619	392	24132	25163
uniform_cost_search	6	9	12	14	41
greedy_h_unmet_goals	6	9	15	18	48
greedy_h_pg_levelsum	6	9	14	17	46
greedy_h_pg_maxlevel	6	9	13	17	45
greedy_h_pg_setlevel	6	9	17	23	55
astar_h_unmet_goals	6	9	12	14	41
astar_h_pg_levelsum	6	9	12	15	42
astar_h_pg_maxlevel	6	9	12	14	41
astar_h_pg_setlevel	6	9	12	14	41